Hyperparameter Optimization Is Deceiving Us, and How to Stop It

A. Feder Cooper[*]
Cornell University
afc78@cornell.edu

Yucheng Lu
Cornell University
yl2967@cornell.edu

Jessica Zosa Forde
Brown University
jforde2@cs.brown.edu

Christopher De Sa
Cornell University
cdesa@cs.cornell.edu

Abstract

Recent empirical work shows that inconsistent results based on choice of hyperparameter optimization (HPO) configuration are a widespread problem in ML research. When comparing two algorithms $\mathcal{J}$ and $\mathcal{K}$, searching one subspace can yield the conclusion that $\mathcal{J}$ outperforms $\mathcal{K}$, whereas searching another can entail the opposite. In short, the way we choose hyperparameters can deceive us. We provide a theoretical complement to this prior work, arguing that, to avoid such deception, the process of drawing conclusions from HPO should be made more rigorous. We call this process *epistemic hyperparameter optimization* (EHPO), and put forth a logical framework to capture its semantics and how it can lead to inconsistent conclusions about performance. Our framework enables us to prove EHPO methods that are guaranteed to be defended against deception, given bounded compute time budget t. We demonstrate our framework's utility by proving and empirically validating a defended variant of random search.

1 Introduction

Machine learning can be informally thought of as a double-loop optimization problem. The *inner loop* is what is typically called *training*: It learns the parameters of some model by running a training algorithm on a training set. This is usually done to minimize some training loss function via an algorithm such as stochastic gradient descent (SGD). Both the inner-loop training algorithm and the model are parameterized by a vector of *hyperparameters* (HPs). Unlike the learned output parameters of a ML model, HPs are inputs provided to the learning algorithm that guide the learning process, such as learning rate and network size. The *outer-loop* optimization problem is to find HPs (from a set of allowable HPs) that result in a trained model that performs the best in expectation on "fresh" examples drawn from the same source as the training set, as measured by some loss or loss approximation. An algorithm that attempts this task is called a *hyperparameter optimization* (HPO) procedure [12, 20].

From this setup comes the natural question: How do we pick the subspace for the HPO procedure to search over? The HPO search space is enormous, suffering from the curse of dimensionality; training, which is also expensive, has to be run for each HP configuration tested. Thus, we have to make hard choices. With limited compute resources, we typically pick a small subspace of possible HPs and perform grid search or random search over that subspace. This involves comparing the empirical performance of the resulting trained models, and then reporting on the model that performs best in

---

[*]Corresponding author: https://cacioepe.pe

terms of a chosen validation metric [20, 37, 41]. For grid search, the grid points are often manually set to values put forth in now-classic papers as good rules-of-thumb concerning, for example, how to set the learning rate [36, 45, 46, 56]. In other words, how we choose which HPs to test can seem rather ad-hoc. We may have a good rationale in mind, but we often elide the details of that rationale on paper; we choose an HPO configuration without explicitly justifying our choice.

Much recent empirical work has critiqued this practice [7, 11, 16, 48, 50, 53, 62, 65]. The authors examine HPO configuration choices in prior work, and find that those choices can have an outsize impact on convergence, correctness, and generalization. They therefore argue that more attention should be paid to the origins of empirical gains in ML, as it is often difficult to tell whether measured improvements are attributable to training or to well-chosen (or lucky) HPs. Yet, this empirical work does not suggest a path forward for formalizing this problem or addressing it theoretically.

To this end, **we argue that the process of drawing conclusions using HPO should itself be an object of study**. Our contribution is to put forward, to the best of our knowledge, the first theoretically-backed characterization for making trustworthy conclusions about algorithm performance using HPO. We model theoretically the following empirically-observed problem: When comparing two algorithms, $\mathcal{J}$ and $\mathcal{K}$, searching one subspace can pick HPs that yield the conclusion that $\mathcal{J}$ outperforms $\mathcal{K}$, whereas searching another can select HPs that entail the opposite result. In short, the way we choose hyperparameters can deceive us—a problem that we call *hyperparameter deception*. We formalize this problem, and prove and empirically validate a defense against it. Importantly, our proven defense does not make any promises about ground-truth algorithm performance; rather, it is guaranteed to avoid the possibility of drawing inconsistent conclusions about algorithm performance within some bounded HPO time budget t. In summary, we:

- Formalize the process of drawing conclusions from HPO (epistemic HPO, Section 3).

- Leverage the flexible semantics of modal logic to construct a framework for reasoning rigorously about 1) uncertainty in epistemic HPO, and 2) how this uncertainty can mislead the conclusions drawn by even the most well-intentioned researchers (Section 4).

- Exercise our logical framework to demonstrate that it naturally suggests defenses with guarantees against being deceived by EHPO, and offer a specific, defended-random-search EHPO (Section 5).

2 Preliminaries: Problem Intuition and Prevalence in ML Research

Principled HPO methods include *grid search* [41] and *random search* [3]. For the former, we perform HPO on a grid of HP-values, constructed by picking a set for each HP and taking the Cartesian product. For the latter, the HP-values are randomly sampled from chosen distributions. Both of these HPO algorithms are parameterized themselves: Grid search requires inputting the spacing between different configuration points in the grid, and random search requires distributions from which to sample. We call these HPO-procedure-input values *hyper-hyperparameters* (hyper-HPs).[2] To make HPO outputs comparable, we also introduce the notion of a *log*:

Definition 1. *A log ℓ records all the choices and measurements made during an HPO run, including the total time T it took to run. It has all necessary information to make the HPO run reproducible.*

A log can be thought of as everything needed to produce a table in a research paper: code, random seed, choice of hyper-HPs, information about the learning task, properties of the learning algorithm, all of the observable results. We formalize all of the randomness in HPO in terms of a random seed r and a pseudo-random number generator (PRNG) G. Given a seed, G deterministically produces a sequence of pseudo-random numbers: all numbers lie in some set $\mathcal{I}$ (typically 64-bit integers), i.e. $r \in \mathcal{I}$ and PRNG $G : \mathcal{I} \to \mathcal{I}^\infty$. With this, we can now define HPO formally:

Definition 2. *An HPO procedure H is a tuple $(H_*, \mathcal{C}, \Lambda, \mathcal{A}, \mathcal{M}, G, X)$ where H_* is a randomized algorithm, $\mathcal{C}$ is a set of allowable hyper-HPs (i.e., allowable configurations for H_*), Λ is a set of allowable HPs (i.e., of HP sets λ), $\mathcal{A}$ is a training algorithm (e.g. SGD), $\mathcal{M}$ is a model (e.g. VGG16), G is a PRNG, and X is some dataset (usually split into train and validation sets). When run, H_* takes as input a hyper-HP configuration $c \in \mathcal{C}$ and a random seed $r \in \mathcal{I}$, then proceeds to run $\mathcal{A}_\lambda$ (on $\mathcal{M}_\lambda$*

---

[2]We provide a glossary of all definitions and symbols for reference at the beginning of the Appendix.

using $G(r)$ and data[3] from X) some number of times for different HPs $\lambda \in \Lambda$. Finally, H_* outputs a tuple (λ^*, ℓ), where λ^* is the HP configuration chosen by HPO and ℓ is the log documenting the run.

Running H is a crucial part of model development. As part of an empirical, scientific procedure, we specify different training algorithms and a learning task, run potentially many HPO passes, and try to make general conclusions about overall algorithm performance. That is, we aim to develop knowledge regarding whether one of the algorithms outperforms the others. However, recent empirical findings indicate that it is actually really challenging to pick hyper-HPs that yield reliable knowledge about general algorithm performance. In fact, it is a surprisingly common occurrence to be able to draw inconsistent conclusions based on our choice of hyper-HPs [11, 16, 48, 65].

An example illustrating the possibility of drawing inconsistent conclusions from HPO. As a first step to studying HPO as a procedure for developing reliable knowledge, we provide an example of how being inadvertently deceived by HPO is a real problem, even in excellent research (we give an additional example in the Appendix).[4] We first reproduce Wilson et al. [72], in which the authors trained VGG16 with different optimizers on CIFAR-10 (Figure 1a). This experiment uses grid search, with a powers-of-2 grid for the learning rate α crossed with the default HPs for Adam. Based on the best-performing HPO per algorithm ($\alpha = 1$), it is reasonable to conclude that non-adaptive methods (e.g., SGD) perform better than adaptive ones (e.g., Adam [42]), as the non-adaptive optimizers demonstrate higher test accuracy.

However, this setting of grid search's hyper-HPs directly informs this particular conclusion; using different hyper-HPs makes it possible to conclude the opposite. Inspired by Choi et al. [11], we perform grid search over a different subspace, tuning both learning rate and Adam's ϵ parameter. Our results entail the logically opposite conclusion: Non-adaptive methods *do not* outperform adaptive ones. Rather, when choosing the HPs that maximize test accuracy, all of the optimizers essentially have equivalent performance (Figure 1b, Appendix). Notably, as we can see from the confidence intervals in Figure 1, **satisfying statistical significance is not sufficient to avoid being deceived about comparative algorithm performance** [73]. Thus, we will require additional tools aside from statistical tests to reason about this, which we discuss in Sections 4 & 5.
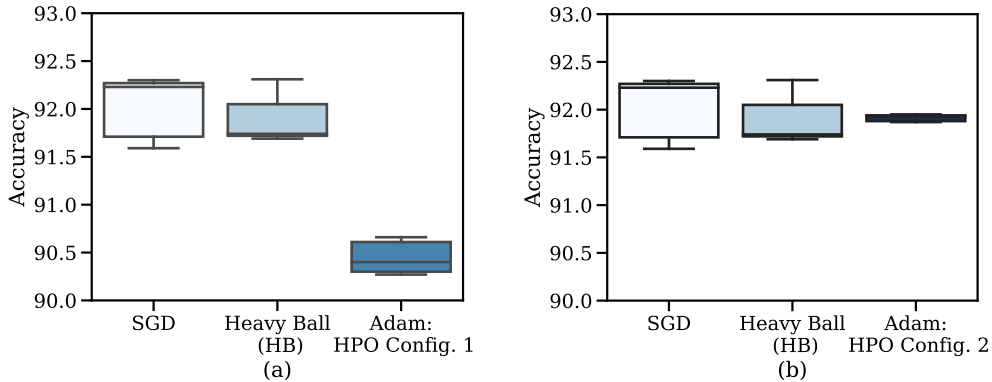

Figure 1: Demonstrating the possibility of drawing inconsistent conclusions from HPO (what we shorthand *hyperparameter deception*) when training VGG16 on CIFAR-10. Each box plot represents a log. In (a), we replicate Wilson et al. [72] and show the best-performing results: One can reasonably conclude that Adam under-performs non-adaptive methods. In (b), we change the HPO search space for Adam, and similarly show the best-performing results: In contradiction, one can reasonably conclude that Adam performs just as well as non-adaptive methods in terms of test accuracy.

This example is not exceptional, or even particularly remarkable, in terms of illustrating the hyperparameter deception problem. We simply chose it for convenience: The experiment does not require highly-specialized ML sub-domain expertise to understand it, and it is arguably broadly familiar, as it very well-cited [72]. However, we emphasize that hyperparameter deception is rather common. Additional examples can be found in numerous empirical studies across ML subfields [7, 11, 16, 49, 50, 53, 60, 65] (Appendix). This work shows that reported results tend to

---

[3]Definition 2 does not preclude cross-validation, as this can be part of H_*. The input dataset X can be split in various ways, as a function of the random seed r.

[4]All code can be found at `https://github.com/pasta41/deception`.

be impressive for the tested hyper-HP configurations, but that modifying HPO can lead to vastly different performance outcomes that entail contradictory conclusions. More generally, it is possible to develop results that are wrong about performance, or else correct about performance but for the wrong reasons (e.g., by picking "lucky" hyperparameters). Neither of these outcomes constitutes reliable knowledge [27, 47]. As scientists, this is disheartening. We want to have confidence in the conclusions we draw from our experiments. We want to trust that we are deriving reliable knowledge about algorithm performance. **In the sections that follow, our aim is to study HPO in this reliable-knowledge sense: We want to develop ways to reason rigorously and confidently about how we derive knowledge from empirical investigations involving HPO.**

3 Epistemic Hyperparameter Optimization

Our discussion in Section 2 shows that applying standard HPO methodologies can be deceptive: Our beliefs about algorithm performance can be controlled by happenstance, wishful thinking, or, even worse, potentially by an adversary trying to trick us with a tampered set of HPO logs. This leaves us in a position where the "knowledge" we derived may not be knowledge at all—since we could have easily (had circumstances been different) concluded the opposite. To address this, we propose that the process of drawing conclusions using HPO should itself be an object of study. We formalize this reasoning process, which we call *epistemic hyperparameter optimization* (EHPO), and we provide an intuition for how EHPO can help us think about the hyperparameter deception problem.

Definition 3. *An **epistemic hyperparameter optimization procedure (EHPO)** is a tuple $(\mathcal{H}, \mathcal{F})$ where $\mathcal{H}$ is a set of HPO procedures H (Definition 2) and $\mathcal{F}$ is a function that maps a set of HPO logs $\mathcal{L}$ (Definition 1) to a set of logical formulas $\mathcal{P}$, i.e. $\mathcal{F}(\mathcal{L}) = \mathcal{P}$. An execution of EHPO involves running each $H \in \mathcal{H}$ some number of times (each run produces a log ℓ), and then evaluating $\mathcal{F}$ on the logs $\mathcal{L}$ produced in order to output the conclusions $\mathcal{F}(\mathcal{L})$ we draw from all of the HPO runs.*

In practice, it is common to run EHPO for two training algorithms, $\mathcal{J}$ and $\mathcal{K}$, and to compare their performance to conclude which is better-suited for the task at hand. $\mathcal{H}$ contains at least one HPO that runs $\mathcal{J}$ and at least one HPO that runs $\mathcal{K}$. The possible conclusions in output $\mathcal{P}$ include $p =$ "$\mathcal{J}$ performs better than $\mathcal{K}$", and $\neg p =$ "$\mathcal{J}$ does not perform better than $\mathcal{K}$". Intuitively, EHPO is deceptive whenever it could produce p and also could (if configured differently or due to randomness) produce $\neg p$. That is, we can be deceived if the EHPO procedure we use to derive knowledge about algorithm performance could entail logically inconsistent results.

Our example in Section 2 is deceptive because using different hyper-HP-configured grid searches for $\mathcal{H}$ could produce contradictory conclusions. We ran two variants of EHPO $(\mathcal{H}, \mathcal{F})$: The first replicated Wilson et al. [72]'s original $\mathcal{H}$ of 3 grid-searches on SGD, HB, and Adam (Figure 1a), and the second used 3 grid-searches with a modified grid search for Adam that also tuned ϵ (Figure 1b). Each EHPO produced a $\mathcal{L}$ with 3 logs. For both, to draw conclusions $\mathcal{F}$ picks the best-performing HP-config per $\mathcal{A}$ and maps them to formulas including "SGD outperforms Adam." From the 3 logs in Figure 1a, we conclude p: "Non-adaptive optimizers outperform adaptive ones"; from the 3 logs in Figure 1b, we conclude $\neg p$: "Non-adaptive methods do not outperform adaptive ones." How can we formally reason about EHPO to avoid this possibility of drawing inconsistent conclusions—to guard against deceiving ourselves about algorithm performance when running EHPO?

Framing an adversary who can deceive us. To begin answering this question, we take inspiration from Descartes' deceptive demon thought experiment (Appendix). We frame the problem in terms of a powerful adversary trying to deceive us—one that can cause us to doubt ourselves and our conclusions. Notably, the demon is not a real adversary; rather, it models a worst-case setting of configurations and randomness that are usually set arbitrarily or by happenstance in EHPO.

Imagine an evil demon who is trying to deceive us about the relative performance of different algorithms via running EHPO. At any time, the demon maintains a set $\mathcal{L}$ of HPO logs, which it can modify either by running an HPO $H \in \mathcal{H}$ with whatever hyper-HPs $c \in \mathcal{C}$ and seed $r \in \mathcal{I}$ it wants (producing a new log ℓ, which it adds to $\mathcal{L}$) or by erasing some of the logs in its set. Eventually, it stops and presents us with $\mathcal{L}$, from which we will draw some conclusions using $\mathcal{F}$, i.e. $\mathcal{F}(\mathcal{L})$. The demon's EHPO could deceive us via the conclusions we draw from the set of logs it produces. For example, $\mathcal{L}$ may lead us to conclude that one algorithm performs better than another, when in fact picking a different set of hyper-HPs could have generated logs that would lead us to conclude differently. We want to be sure that we will not be deceived by any logs the demon ***could*** produce. Of

course, this intuitive definition is lacking: It is not clear what is meant by ***could***. Our contribution in the sections that follow is to pin down a formal, reasonable definition of ***could*** in this context, so that we can suggest an EHPO procedure that can defend against such a maximally powerful adversary. We intentionally imagine such a powerful adversary because, if we can defend against it, then we will also be defended against weaker or accidental deception.

4   A Logic for Reasoning about EHPO

The informal notion of ***could*** established above encompasses numerous sources of uncertainty. There is the time to run EHPO and the choices of random seed, algorithms to compare, HPO procedures, hyper-HPs, and learning task. Then, once we have completed EHPO and have a set of logs, we have to digest those logs into logical formulas from which we base our conclusions. This introduces more uncertainty, as we need to reason about whether we believe those conclusions or not. Our formalization needs to capture all of these sources of uncertainty, and needs to be sufficiently expressive to capture how they could combine to cause us to believe deceptive conclusions. It needs to be expansive enough to handle the common case—of a well-intentioned researcher with limited resources making potentially incorrect conclusions—and the rarer, worst case—of gaming results.

Why not statistics? As the common toolkit in ML, statistics might seem like the right choice for modeling all this uncertainty. However, statistics is great for reasoning about uncertainty that is *quantifiable*. For this problem, not all of the sources of uncertainty are easily quantifiable. In particular, it is very difficult to quantify the different hyper-HP possibilities. It is not reasonable to model hyper-HP selection as a random process; we do not sample from a distribution and, even if we wanted to, it is not clear how we would pick the distribution from which to sample. Moreover, as we saw in our example in Section 2, testing for statistical significance is not sufficient to prevent deception. While the results under consideration may be statistically significant, they can still fail to prevent the possibility of yielding inconsistent conclusions. For this reason, when it comes to deception, statistical significance can even give us false confidence in the conclusions we draw.

Why modal logic? Modal logic is the standard mathematical tool for formalizing reasoning about uncertainty [10, 18]—for formalizing the thus far informal notion of what the demon ***could*** bring about running EHPO. It is meant precisely for dealing with different types of uncertainty, particularly uncertainty that is difficult to quantify, and has been successfully employed for decades in AI [6, 30, 31], programming languages [13, 44, 58], and distributed systems [21, 32, 55]. In each of these computer science fields, modal logic's flexible semantics has been indispensable for writing proofs about higher-level specifications with multiple sources of not-precisely-quantifiable, lower-level uncertainty. For example, in distributed computing, it lets us write proofs about overall system correctness, abstracting away from the specific non-determinism introduced by each lower-level computing process [21]. Analogously, modal logic can capture the uncertainty in EHPO without being prescriptive about particular hyper-HP choices. Our notion of correctness, which we want to reason about and guarantee, is not being deceived. Therefore, while modal logic may be an atypical choice for ML, it comes with a huge payoff. By constructing the right semantics, we can capture all the sources of uncertainty described above and we can write simple proofs about whether we can be deceived by the EHPO we run. In Section 5, it is this formalization that ultimately enables us to naturally suggest a defense against being deceived.

4.1   Introducing our logic: syntax and semantics overview

Modal logic inherits the tools of more-familiar propositional logic and adds two operators: $\Diamond$ to represent *possibility* and $\Box$ to represent *necessity*. These operators enable reasoning about *possible worlds*—a semantics for representing how the world *is* or *could be*, making modal logic the natural choice to express the "could" intuition from Section 3. The well-formed formulas ϕ of modal logic are given recursively in Backus-Naur form, where P is any atomic proposition:

$$\phi := P \mid \neg\phi \mid \phi \wedge \phi \mid \Diamond\phi$$

$\Diamond p$ reads, "It is possible that p."; p is true at *some* possible world, which we *could* reach (Appendix). Note that $\Box$ is syntactic sugar, with $\Box p \equiv \neg\Diamond\neg p$. Similarly, "or" has $p \vee q \equiv \neg(\neg p \wedge \neg q)$ and "implies" has $p \rightarrow q \equiv \neg p \vee q$. The axioms of modal logic are as follows:

$\vdash Q \rightarrow \Box Q$ *(necessitation).* $\Box(Q \rightarrow R) \rightarrow (\Box Q \rightarrow \Box R)$ *(distribution).*

where Q and R are any formula, and $\vdash Q$ means Q is a theorem of propositional logic. We can now provide the syntax and an intuitive notion of the semantics of our logic for reasoning about deception.

Syntax. Our logic requires an extension of standard modal logic. We need *two* modal operators to reckon with two overarching modalities: the possible results of the demon running EHPO ($\Diamond_t$) and our beliefs about conclusions from those results ($\mathcal{B}$). Combining these modalities yields well-formed formulas ψ where, for any atomic proposition P and any positive real t,

$$\psi := P \mid \neg\psi \mid \psi \wedge \psi \mid \Diamond_t\psi \mid \mathcal{B}\psi$$

Note the EHPO modal operator here is *indexed*: $\Diamond_t$ captures "how possible" ($\Diamond$) something is, quantified by the compute capabilities of the demon (t) [6, 18, 33].

Semantics intuition. We suppose that an EHPO user has in mind some atomic propositions (propositions of the background logic unrelated to possibility or belief, such as "the best-performing log for $\mathcal{J}$ has lower loss than the best-performing log for $\mathcal{K}$") with semantics that are already defined. $\wedge$ and $\neg$ inherit their semantics from ordinary propositional logic, which can combine propositions to form formulas. A set of EHPO logs $\mathcal{L}$ (Definition 1) can be digested into such logical formulas. That is, we define our semantics using logs $\mathcal{L}$ as models over formulas p: $\mathcal{L} \models p$, which reads "$\mathcal{L}$ models p", means that p is true for the set of logs $\mathcal{L}$. We will extend this intuition to give semantics for possibility $\Diamond_t$ (Section 4.2) and belief $\mathcal{B}$ (Section 4.3), culminating in a tool that lets us reason about whether or not EHPO can deceive us by possibly yielding inconsistent conclusions (Section 4.4).

Using our concrete example to ground us. To clarify our presentation below, we will map our semantics to the example from Section 2, providing an informal intuition before formal definitions.

4.2 Expressing the possible outcomes of EHPO using $\Diamond_t$

Our formalization for possible EHPO is based on the demon of Section 3. Recall, the demon models a worst-case scenario. In practice, we deal with the easier case of well-intentioned ML researchers. The notion of possibility we define here gives limits on what possible world a demon *with bounded EHPO time* could reliably bring about. We first define a *strategy* the demon can execute for EHPO:

Definition 4. *A randomized* **strategy** *σ is a function that specifies which action the demon will take. Given $\mathcal{L}$, its current set of logs, $\sigma(\mathcal{L})$ gives a distribution over concrete actions, where each action is either 1) running a new H with its choice of hyper-HPs c and seed r 2) erasing some logs, or 3) returning. We let Σ denote the set of all such strategies.*

The demon we model controls the hyper-HPs c and the random seed r, but importantly does *not* fully control the PRNG G. From the adversary's perspective, for a strategy σ to be reliable it must succeed regardless of the specific G. Informally, the demon cannot hack the PRNG.[5]

Informally, we now want to *execute a strategy* to bring about a particular outcome p. In Section 2, our good-faith strategy was simple: We ran each H with its own hyper-HPs and random seed, then returned. The demon is trickier: It is adopting a strategy to try to bring about a deceptive outcome. **Formally**, we model the demon executing strategy σ on logs $\mathcal{L}$ with a PRNG unknown to the demon as follows. Let $\mathcal{G}$ denote the distribution over PRNGs $G : \mathcal{I} \to \mathcal{I}^\infty$, in which all number sequence elements are drawn independently and uniformly from $\mathcal{I}$ (recall, $\mathcal{I}$ is typically the 64-bit integers). First, draw G from $\mathcal{G}$, conditioned on G being consistent with all the runs in $\mathcal{L}$.[6] The demon then performs a random action drawn from $\sigma(\mathcal{L})$, using G as the PRNG when running a new HPO H, and continues—updating the working set of logs $\mathcal{L}$ as it goes—until the "return" action is chosen.

Using this process, we define what outcomes p the demon can reliably bring about (i.e., what is possible, $\Diamond$) in the EHPO output logs $\mathcal{L}$ by running this random strategy σ in bounded time t. **Informally**, $\Diamond_t p$ means that an adversary could adopt a strategy σ that is guaranteed to cause the desired outcome p to be the case while taking time at most t in expectation. In Section 2, where p is "Non-adaptive methods outperform adaptive ones", Figure 1a shows $\Diamond_t p$. **Formally**,

Definition 5. *Let $\sigma[\mathcal{L}]$ denote the logs output from executing strategy σ on logs $\mathcal{L}$, and let $\tau_\sigma(\mathcal{L})$ denote the total time spent during execution. $\tau_\sigma(\mathcal{L})$ is equivalent to the sum of the times T it took*

---

[5]We do not consider adversaries that can directly control how data is ordered and submitted to the algorithms under evaluation. This distinction shows that our logical construction non-trivial: We are able to defend against strong adversaries that can game the output of EHPO, which is separate from cheating by hacking the PRNG.

[6]i.e., All random events recorded in $\mathcal{L}$ should agree with the corresponding random numbers produced by G.

each HPO procedure $H \in \mathcal{H}$ executed in strategy σ to run. Note that both $\sigma[\mathcal{L}]$ and $\tau_\sigma(\mathcal{L})$ are random variables, as a function of the randomness of selecting G and the actions sampled from $\sigma(\mathcal{L})$. For any formula p and any $t \in \mathbb{R}_{>0}$, we say $\mathcal{L} \models \Diamond_t p$, i.e. "$\mathcal{L}$ models that it is possible p in time t," if

$$\text{there exists a strategy } \sigma \in \Sigma, \text{ such that } \quad \mathbb{P}(\sigma[\mathcal{L}] \models p) = 1 \text{ and } \mathbb{E}[\tau_\sigma(\mathcal{L})] \leq t.$$

We will usually choose t to be an upper bound on what is considered a reasonable amount of time to run EHPO. It does not make sense for t to be unbounded, since this corresponds to the unrealistic setting of having infinite compute time to perform HPO runs. We model our budget in terms of time; however, we could use this setup to reason about other monotonically increasing resource costs, such as energy usage. Our indexed modal logic inherits many axioms of modal logic, with indexes added (Appendix), e.g.:

$$\vdash (p \rightarrow q) \rightarrow (\Diamond_t p \rightarrow \Diamond_t q) \quad \textit{(necess. + distribution)} \qquad p \rightarrow \Diamond_t p \quad \textit{(reflexivity)}$$
$$\Diamond_t \Diamond_s p \rightarrow \Diamond_{t+s} p \qquad\qquad \textit{(transitivity)} \qquad \Diamond_s \Box_t p \rightarrow \Box_t p \quad \textit{(symmetry)}$$
$$\Diamond_t (p \wedge q) \rightarrow (\Diamond_t p \wedge \Diamond_t q) \qquad \textit{(dist. over $\wedge$)},$$

To summarize: The demon knows all possible hyper-HPs; it can pick whichever ones it wants to run EHPO within a bounded time budget t to realize the outcome p it wants. That is, if with some probability the demon can deceive us in some amount of time, then the demon can reliably deceive us with any larger time budget: If the demon fails to produce a deceptive result, it can use the strategy of just re-running until it yields the result it desires. Since $\Diamond_t$ models the worst-case all-powerful demon, it can also model any weaker EHPO user with time budget t.

4.3 Expressing how we draw conclusions using $\mathcal{B}$

We employ the modal operator $\mathcal{B}$ from the logic of belief[7] to model ourselves as an observer who believes in the truth of the conclusions drawn from running EHPO. $\mathcal{B}p$ reads "It is concluded that p." For example, when comparing the performance of two algorithms for a task, p could be "$\mathcal{J}$ is better than $\mathcal{K}$" and thus $\mathcal{B}p$ would be understood as, "It is concluded that $\mathcal{J}$ is better than $\mathcal{K}$."

We model ourselves as a consistent *Type 1* reasoner [67]. **Informally**, this means we believe all propositional tautologies (necessitation), our belief distributes over implication (distribution), and we do not derive contradictions (consistency). We do not require completeness: We allow the possibility of not concluding anything about p (i.e., neither $\mathcal{B}p$ nor $\mathcal{B}\neg p$). **Formally**, for any formulas p and q,

$$\vdash p \rightarrow \mathcal{B}p \quad \textit{(necess.)}; \quad \mathcal{B}(p \rightarrow q) \rightarrow (\mathcal{B}p \rightarrow \mathcal{B}q) \quad \textit{(dist.)}; \quad \neg(\mathcal{B}p \wedge \mathcal{B}\neg p) \quad \textit{(consistency)}.$$

To understand our belief semantics, recall that EHPO includes a function $\mathcal{F}$, which maps a set of output logs $\mathcal{L}$ to our conclusions (i.e., $\mathcal{F}(\mathcal{L}) = \mathcal{P}$ is our set of conclusions). **Informally**, when our conclusion set $\mathcal{F}(\mathcal{L})$ contains a formula p, we say the set of logs $\mathcal{L}$ models our belief $\mathcal{B}$ in that formula p. In Section 2, the logs of Figure 1a model $\mathcal{B}p$ and the logs of Figure 1b model $\mathcal{B}\neg p$. **Formally**,

Definition 6. *For any formula p, we say $\mathcal{L} \models \mathcal{B}p$, "$\mathcal{L}$ models our belief in p", if $p \in \mathcal{F}(\mathcal{L})$.*

Note we constrain what $\mathcal{F}$ can output. For a reasonable notion of belief, $\mathcal{F}$ must model the consistent *Type 1* reasoner axioms above. Otherwise, deception aside, $\mathcal{F}$ is an unreasonable way to draw conclusions, since it is not even compatible with our belief logic.

4.4 Expressing hyperparameter deception

So far we have defined the semantics of our two separate modal operators, $\Diamond_t$ and $\mathcal{B}$. We now begin to reveal the benefit of using modal logic for our formalization. These operators can interact to formally express what we informally illustrated in Section 2: a notion of hyperparameter deception. It is a well-known result that we can combine modal logics [61] (Appendix). We do so to define an axiom that, if satisfied, guarantees EHPO will not be able to deceive us. For any formula p,

$$\neg (\Diamond_t \mathcal{B}p \wedge \Diamond_t \mathcal{B}\neg p) \qquad\qquad \textit{(t-non-deceptive)}.$$

Informally, our running example can be considered a proof by exhibition: It violates this axiom because Figure 1a's logs model $\Diamond_t \mathcal{B}p$ and Figure 1b's logs model $\Diamond_t \mathcal{B}\neg p$. That is, $\Diamond_t \mathcal{B}p \wedge \Diamond_t \mathcal{B}\neg p$ using grid search for this task.

---

[7]$\mathcal{B}$ is syntactically analogous to the $\Box$ modal operator in standard modal logic [35, 63, 69] (Appendix).

For the worst-case, *t-non-deceptiveness* expresses the following: **If there exists a strategy σ by which the demon could get us to conclude p in t expected time, then there can exist no t-time strategy by which the demon could have gotten us to believe $\neg p$.** To make this concrete, suppose our t-non-deceptive axiom holds for an EHPO method that results in p. Intuitively, given a maximum reasonable time budget t, if there is no adversary that can consistently control whether we believe p or its negation when running that EHPO, then the EHPO is defended against deception. Conversely, if an adversary could consistently control our conclusions, then the EHPO is potentially gameable. That is, if our t-non-deceptive axiom does not hold (i.e., we can be deceived, $\Diamond_t \mathcal{B} p \wedge \Diamond_t \mathcal{B} \neg p$), then even if we conclude p after running EHPO, we cannot claim to *know p*. Our belief as to the truth-value of p could be under the complete control of an adversary—or just a result of happenstance.

To summarize: An EHPO is t-non-deceptive if it satisfies all of the axioms above. Our example in Section 2 is t-deceptive because the axioms do not hold. The semantics of these axioms capture all of the possible uncertainty from the process of drawing conclusions from EHPO–and how that uncertainty can combine to cause us to believe t-deceptive conclusions.

5 Constructing Defended EHPO

Now that we have a formal notion of what it means for EHPO to be (non)-deceptive, **we can write proofs about what it means for an EHPO method to be guaranteed to be deception-free**. Importantly, these proofs will increase our confidence that our conclusions from EHPO are not due to the happenstance of picking a particular set of hyper-HPs.

To talk about defenses, we need to understand what it means to construct a "defended reasoner." In other words, for an EHPO $(\mathcal{H}, \mathcal{F})$, we need $\mathcal{F}$ to yield conclusions that we can defend against deception. Recall from Definition 6 that logs $\mathcal{L}$ model our belief in a formula p, i.e. $\mathcal{L} \models \mathcal{B} p \equiv p \in \mathcal{F}(\mathcal{L})$. With this in mind, we begin by supposing we have a naive EHPO $(\mathcal{H}, \mathcal{F}_n)$ featuring a naive reasoner $\mathcal{B}_n$ with corresponding belief function $\mathcal{F}_n$. We want to construct a new "defended reasoner" $\mathcal{B}_*$ that has a "skeptical" belief function $\mathcal{F}_*$. $\mathcal{F}_*$ should weaken the conclusions of $\mathcal{F}_n$ (i.e., $\mathcal{F}_*(\mathcal{L}) \subseteq \mathcal{F}_n(\mathcal{L})$ for any $\mathcal{L}$) and result in an EHPO $(\mathcal{H}, \mathcal{F}_*)$ that is guaranteed to be t-non-deceptive. In other words, defended reasoner $\mathcal{B}_*$ never concludes more than the naive reasoner $\mathcal{B}_n$. **Informally**, a straightforward way to do this is to have $\mathcal{B}_*$ conclude p only if both the naive $\mathcal{B}_n$ would have concluded p, and it is impossible for an adversary to get $\mathcal{B}_n$ to conclude $\neg p$ in time t. **Formally**, construct $\mathcal{B}_*$ such that for any p, $\mathcal{B}_* p \equiv \mathcal{B}_n p \wedge \neg \Diamond_t \mathcal{B}_n \neg p$ (1).

Directly from our axioms (Section 4), we can now prove $\mathcal{B}_*$ is defended. We will suppose it is possible for $\mathcal{B}_*$ to be deceived, demonstrate a contradiction, and thereby guarantee that $\mathcal{B}_*$ is t-non-deceptive. Suppose $\mathcal{B}_*$ can be deceived in time t, i.e. $\Diamond_t \mathcal{B}_* p \wedge \Diamond_t \mathcal{B}_* \neg p$ is True. Starting with the left, $\Diamond_t \mathcal{B}_* p$:

		Rule
$\Diamond_t \mathcal{B}_* p$	$\equiv$ $\Diamond_t (\mathcal{B}_n p \wedge \neg \Diamond_t \mathcal{B}_n \neg p)$	Applying $\Diamond_t$ to the definition of $\mathcal{B}_* p$ (1)
	$\rightarrow$ $\Diamond_t (\neg \Diamond_t \mathcal{B}_n \neg p)$	Reducing a conjunction to either of its terms: $(a \wedge b) \rightarrow b$
	$\rightarrow$ $\neg \Diamond_t \mathcal{B}_n \neg p$	Symmetry; dropping all but the right-most operator: $\Diamond_t(\Diamond_t a) \rightarrow \Diamond_t a$

We then pause to apply our axioms to the right side of the conjunction, $\Diamond_t \mathcal{B}_* \neg p$:

		Rule
$\Diamond_t \mathcal{B}_* \neg p$	$\equiv$ $\Diamond_t (\mathcal{B}_n \neg p \wedge \neg \Diamond_t \mathcal{B}_n p)$	Applying $\Diamond_t$ to the definition of $\mathcal{B}_* \neg p$ (1)
	$\rightarrow$ $\Diamond_t \mathcal{B}_n \neg p \wedge \Diamond_t \neg \Diamond_t \mathcal{B}_n p$	Distributing $\Diamond_t$ over $\wedge$: $\Diamond_t(a \wedge b) \rightarrow (\Diamond_t a \wedge \Diamond_t b)$
	$\rightarrow$ $\Diamond_t \mathcal{B}_n \neg p$	Reducing a conjunction to either of its terms: $(a \wedge b) \rightarrow a$

We now bring both sides of the conjunction back together: $\Diamond_t \mathcal{B}_* p \wedge \Diamond_t \mathcal{B}_* \neg p \equiv \neg \Diamond_t \mathcal{B}_n \neg p \wedge \Diamond_t \mathcal{B}_n \neg p$. The right-hand side is of the form $\neg a \wedge a$, which must be False. This contradicts our initial assumption that $\mathcal{B}_*$ is t-deceptive (i.e., $\Diamond_t \mathcal{B}_* p \wedge \Diamond_t \mathcal{B}_* \neg p$ is True). Therefore, $\mathcal{B}_*$ is t-non-deceptive.

This example illustrates the power of our choice of formalization. In just a few lines of simple logic, we can validate defenses against deception. **This analysis shows that a t-defended reasoner $\mathcal{B}_*$ is always possible**, and it does so without needing to refer to the particular underlying semantics of an

EHPO. However, we intend this example to only be illustrative, as it may not be practical to compute $\mathcal{B}_*$ as defined in (1) if we cannot easily evaluate whether $\Diamond_t \mathcal{B}_n \neg p$. We next suggest a concrete EHPO with a defended $\mathcal{B}_*$, and show how deception can be avoided in our Section 2 example by using this EHPO instead of grid search.

A defended random search EHPO. Random search takes two hyper-HPs, a distribution μ over the HP space and a number of trials $K \in \mathbb{N}$ to run. HPO consists of K independent trials of training algorithms $\mathcal{A}_{\lambda_1}, \mathcal{A}_{\lambda_2}, \ldots, \mathcal{A}_{\lambda_K}$, where the HPs λ_k are independently drawn from μ, taking expected time proportional to K. When drawing conclusions, we usually look at the "best" run for each algorithm. For simplicity, we suppose there is only one algorithm, $\mathcal{A}$. We bound how much the choice of hyper-HPs can affect the HPs, and define a defended EHPO based on a variant of random search.

Definition 7. *Suppose that we are given a naive EHPO procedure $(\{H\}, \mathcal{F}_n)$, in which H is random search and is the only HPO in our EHPO, and $\mathcal{F}_n$ is a "naive" belief function associated with a naive reasoner $\mathcal{B}_n$. For any $K, R \in \mathbb{N}$, we define the "(K, R)-defended" belief function $\mathcal{F}_*$ for a skeptical reasoner $\mathcal{B}_*$ as the following conclusion-drawing procedure. First, $\mathcal{F}_*$ only makes conclusion set $\mathcal{P}_*$ from a single log $\hat{\ell}$ with $K * R$ trials; otherwise, it concludes nothing, outputting $\emptyset$. Second, $\mathcal{F}_*$ splits the single $\hat{\ell}$ into R logs $\ell_1, \ell_2, \ldots, \ell_R$, each containing K independent-random-search trials.[8] Finally, $\mathcal{F}_*$ outputs the intersection of what the naive reasoner would have output on each log ℓ_i,*

$$\mathcal{F}_*(\{\hat{\ell}\}) = \mathcal{P}_* \equiv \mathcal{F}_n(\{\ell_1\}) \cap \mathcal{F}_n(\{\ell_2\}) \cap \cdots \cap \mathcal{F}_n(\{\ell_R\}).$$

Equivalently, $\{\hat{\ell}\} \models \mathcal{B}_* p$ only if $\{\ell_i\} \models \mathcal{B}_n p$ for all i.

Informally, to draw a conclusion using this EHPO, $\mathcal{B}_*$ splits a random-search-trial log of size $K * R$ into R groups of K-trial logs, passing each K-trial log to one of an ensemble of R naive reasoners $\mathcal{B}_n$. $\mathcal{B}_*$ only concludes p if all R naive reasoners unanimously agree on p. We can guarantee this EHPO to be t-non-deceptive by assuming a bound on how much the hyper-HPs can affect the HPs.

Theorem 1. *Suppose that the set of allowable hyper-HPs $\mathcal{C}$ of H is constrained, such that any two allowable random-search distributions μ and ν have Renyi-∞-divergence at most a constant, i.e. $D_\infty(\mu \| \nu) \leq \gamma$. The (K, R)-defended random-search EHPO of Definition 7 is guaranteed to be t-non-deceptive if we set $R \geq \sqrt{t \exp(\gamma K)/K} = O(\sqrt{t})$.*

We prove Theorem 1 in the Appendix. This result shows that our defense is *actually* a defense, and moreover it defends with a log size $K * R$—and compute requirement for good-faith EHPO—that scales sublinearly in t. A good-faith actor can, in sublinear-in-t time, produce a log (of length $K * R$) that will allow our t-non-deceptive reasoner to reach conclusions. This means that we defend against adversaries with much larger compute budgets than are expected from good-faith actors.

Validating our defense empirically and selecting hyper-HPs. Any defense ultimately depends on the hyper-HPs it uses. Thus, we should have a reasonable belief that choosing differently would not have led an opposite conclusion. We therefore run a two-phased search [11, 34, 59], repeating our VGG16-CIFAR10 experiment from Section 2. First, we run a coarse-grained, dynamic protocol to find reasonable hyper-HPs for Adam's ϵ; second, we use those hyper-HPs to run our defended random search. We start with a distribution to search over ϵ, and note that the performance is best on the high end. We change the hyper-HPs, shifting the distribution until Adam's performance starts to degrade, and use the resulting hyper-HPs ($\epsilon \in [10^{10}, 10^{12}]$) to run our defense (Appendix).

We now run a modified version of our defended EHPO in Definition 7, described in Algorithm 1, with $K * R = 600$ (200 logs for each optimizer). Using a budget of $M = 10000$ iterations, we subsample $\kappa = 11$ logs and pass them to an ensemble of κ naive reasoners $\mathcal{B}_n$. We use κ logs, relaxing the requirement of using all $K * R$ logs in Definition 7, for efficiency. Each iteration m concludes the majority conclusion of the κ-sized $\mathcal{B}_n$ ensemble. This is why we set κ to an odd number—to avoid ties. $\mathcal{B}_*$ draws conclusions based on the results of the M-majority conclusions. That is, we further relax the requirements of Definition 7: Instead of requiring unanimity, $\mathcal{B}_*$ only requires agreement on the truth-value of p for a fractional subset of M. We set this fraction using parameter $\delta \in [0, 1]$, where δ controls how skeptical our defended reasoner $\mathcal{B}_*$ is (lower δ corresponding to more skepticism). $\mathcal{B}_*$ concludes p when at least $(1 - \delta)$ of our M subsampled runs concluded p. When this threshold is not met, $\mathcal{B}_*$ remains skeptical and concludes nothing. We summarize our final results in Table 1, and

---

[8]This is not generally allowable. $\mathcal{F}_*$ can do this because random-search logs contain interchangeable trials.

provide complete results in the Appendix. Given how similar the optimizers all perform on this task (similar to Figure 1), being more skeptical increases the likelihood that we do not conclude anything.

Algorithm 1 Defense with Random Search

Require: Set of $K * R$ random-search logs $\{\mathcal{L}_i\}_{i=1}^{KR}$,
  defense subsampling budget M,
  criterion constant δ, subsample size κ.
1: **for** $m = 1, \cdots, M$ **do**
2:   Subsample κ logs: $\{\mathcal{L}_i\}_{i=1}^{\kappa} \sim \{\mathcal{L}_i\}_{i=1}^{KR}$.
3:   Obtain conclusions $\{\mathcal{P}_i\}_{i=1}^{\kappa}$ from $\{\mathcal{L}_i\}_{i=1}^{\kappa}$.
4:   Obtain output conclusion for m:
       $\mathcal{P}^{(m)} \leftarrow \text{Majority}(\{\mathcal{P}_i\}_{i=1}^{\kappa})$
5: **end for**
6: **if** $\exists p$ s.t. $\geq (1-\delta)M$ of $\{\mathcal{P}^{(m)}\}_{i=1}^{M}$ conclude p **then**
7:   Conclude p.
8: **else**
9:   Conclude nothing.
10: **end if**

Table 1: Results from repeating our Section 2 experiment, using Algorithm 1 instead of grid search. p = "Non-adaptive optimizers (SGD and HB) perform better than the adaptive optimizer Adam".

	p	$\neg p$	$1-\delta$	Conclude
SGD vs. Adam	0.213	0.788	0.75	$\neg p$
			0.8	Nothing
			0.9	Nothing
HB vs. Adam	0.168	0.832	0.75	$\neg p$
			0.8	$\neg p$
			0.9	Nothing

6 Conclusion and Practical Takeaways

Much recent empirical work illustrates that it is easy to draw inconsistent conclusions from HPO [7, 11, 16, 49, 50, 53, 60, 65]. We call this problem *hyperparameter deception* and, to derive a defense, **argue that the process of drawing conclusions using HPO should itself be an object of study**. Taking inspiration from Descartes' demon, we formalize a logic for studying an epistemic HPO procedure. The demon can run any number of reproducible HPO passes to try to get us to believe a particular notion about algorithm performance. Our formalization enables us to not believe deceptive notions: It naturally suggests how to guarantee that an EHPO is defended against deception. We offer recommendations to avoid hyperparameter deception in practice (we expand on this in the Appendix):

- **Researchers should construct their own notion of skepticism $\mathcal{B}_*$, appropriate to their specific task.** There is no one-size-fits-all defense solution. Our results are ***broad insights*** about defended EHPO: A defended EHPO is ***always possible***, but finding an efficient one will depend on the task.

- **Researchers should make explicit how they choose hyper-HPs.** What is reasonable is ultimately a function of what the ML community accepts. Being explicit, rather than eliding hyper-HP choices, is essential for helping decide what is reasonable. As a heuristic, we recommend setting hyper-HPs such that they include HPs for which the optimizers' performance starts to degrade, as we do above.

- **Avoiding hyperparameter deception is just as important as reproducibility**. We have shown that reproducibility [7, 29, 34, 57, 64] is only part of the story for ensuring reliability. While necessary for guarding against brittle findings, it is not sufficient. We can replicate results—even statistically significant ones—that suggest conclusions that are altogether wrong.

More generally, our work is a call to researchers to reason more rigorously about their beliefs concerning algorithm performance. In relation to EHPO, this is akin to challenging researchers to reify their notion of $\mathcal{B}$—to justify their belief in their conclusions from the HPO. Such epistemic rigor concerning drawing conclusions from empirical studies has a long history in more mature branches of science and computing, including evolutionary biology [28], statistics [25, 26], programming languages [54], and computer systems [23] (Appendix). We believe that applying similar rigor will contribute significantly to the ongoing effort of making ML more robust and reliable.

Acknowledgements

References

[1] Mark Van Atten. *On Brouwer*. Cengage Learning, 2004.

[2] Alexandru Baltag and Sonja Smets. Probabilistic dynamic belief revision. *Synthese*, 165(2): 179–202, 2008.

[3] James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.*, 13:281–305, February 2012. ISSN 1532-4435.

[4] James S. Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for Hyper-Parameter Optimization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2546–2554. Curran Associates, Inc., 2011.

[5] Guram Bezhanishvili and Wesley H. Holliday. A semantic hierarchy for intuitionistic logic. *Indagationes Mathematicae*, 30(3):403 – 469, 2019. ISSN 0019-3577.

[6] Patrick Blackburn, Johan F. A. K. van Benthem, and Frank Wolter. *Handbook of Modal Logic*, volume 3. Elsevier Science Inc., USA, 2006. ISBN 0444516905.

[7] Xavier Bouthillier, César Laurent, and Pascal Vincent. Unreproducible Research is Reproducible. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 725–734. PMLR, 09–15 Jun 2019.

[8] Roberto Calandra, Ignasi Clavera Gilaberte, Frank Hutter, Joaquin Vanschoren, and Jane Wang. Meta-learning. Meta-Learning Workshop at NeurIPS 2020, 2020.

[9] Jimena Canales. *Bedeviled: A Shadow History of Demons in Science*. Princeton University Press, Princeton, NJ, USA, 2020.

[10] Brian F. Chellas. *Modal Logic - An Introduction*. Cambridge University Press, 1980.

[11] Dami Choi, Christopher J. Shallue, Zachary Nado, Jaehoon Lee, Chris J. Maddison, and George E. Dahl. On Empirical Comparisons of Optimizers for Deep Learning, 2019.

[12] Marc Claesen and Bart De Moor. Hyperparameter Search in Machine Learning, 2015.

[13] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. *ACM Trans. Program. Lang. Syst.*, 8(2):244–263, April 1986. ISSN 0164-0925.

[14] A. Feder Cooper and Ellen Abrams. Emergent Unfairness in Algorithmic Fairness-Accuracy Trade-Off Research. In *Artificial Intelligence, Ethics, and Society (AIES)*, 2021.

[15] René Descartes. *Discourse on Method and Meditations on First Philosophy*. Hackett Publishing Company, Inc., Translator Donald A. Cress, 4th edition, 1998. Meditation One: Concerning Those Things That Can Be Called into Doubt.

[16] Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. Show Your Work: Improved Reporting of Experimental Results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2185–2194, Hong Kong, China, November 2019. Association for Computational Linguistics.

[17] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20:1–21, 2018.

[18] E. Allen Emerson. *Temporal and Modal Logic*, page 995–1072. MIT Press, Cambridge, MA, USA, 1991. ISBN 0444880747.

[19] Eduardo Fermé and Sven Ove Hansson. AGM 25 Years: Twenty-Five Years of Research in Belief Change. *Journal of Philosophical Logic*, 40:295–331, April 2011.

[20] Matthias Feurer and Frank Hutter. Hyperparameter Optimization. In Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors, *Automated Machine Learning: Methods, Systems, Challenges*, pages 3–33. Springer International Publishing, 2019.

[21] Michael J. Fischer and Lenore D. Zuck. Reasoning about uncertainty in fault-tolerant distributed systems. In M. Joseph, editor, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, pages 142–158, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.

[22] Jessica Zosa Forde, A. Feder Cooper, Kweku Kwegyir-Aggrey, Chris De Sa, and Michael Littman. Model Selection's Disparate Impact in Real-World Deep Learning Applications, 2021. ICML 2021 Science of Deep Learning Workshop.

[23] Batya Friedman and Helen Nissenbaum. Bias in Computer Systems. *ACM Trans. Inf. Syst.*, 14 (3):330–347, July 1996. ISSN 1046-8188.

[24] James Garson. Modal Logic. In *The Stanford Encyclopedia of Philosophy*. Fall 2018 Edition, Edward N. Zalta (ed.), 2018.

[25] A. Gelman and Eric Loken. The garden of forking paths: Why multiple comparisons can be a problem, even when there is no "fishing expedition" or "p-hacking" and the research hypothesis was posited ahead of time, 2019.

[26] Andrew Gelman and Eric Loken. The statistical crisis in science: data-dependent analysis—a "garden of forking paths"–explains why many statistically significant comparisons don't hold up. *American Scientist*, 102(6):460–466, 2014.

[27] Edmund L. Gettier. Is Justified True Belief Knowledge? *Analysis*, 23(6):121–123, 06 1963.

[28] Stephen Jay Gould. *The Mismeasure of Man*. Norton, New York, 1981.

[29] Odd Erik Gundersen and Sigbjørn Kjensmo. State of the Art: Reproducibility in Artificial Intelligence. In *AAAI*, 2018.

[30] Joseph Y. Halpern. *Reasoning about Uncertainty*. The MIT Press, 2 edition, 2017.

[31] Joseph Y. Halpern and Y. Shoham. A propositional modal logic of time intervals. In *Journal of the ACM*, 1991.

[32] Chris Hawblitzel, Jon Howell, Manos Kapritsos, Jacob R. Lorch, Bryan Parno, Michael L. Roberts, Srinath Setty, and Brian Zill. IronFleet: Proving Practical Distributed Systems Correct. In *Proceedings of the 25th Symposium on Operating Systems Principles*, SOSP '15, page 1–17, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450338349.

[33] Aviad Heifeitz and Philippe Mongin. The Modal Logic of Probability. In *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 175–185, July 1998.

[34] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep Reinforcement Learning that Matters. In *Thirty-Second AAAI Conference On Artificial Intelligence*, 2018.

[35] Jaakko Hintikka. *Knowledge and Belief*. Cornell University Press, 1962.

[36] Geoffrey E. Hinton. A Practical Guide to Training Restricted Boltzmann Machines. In Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade: Second Edition*, pages 599–619. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[37] Chihwei Hsu, Chihchung Chang, and ChihJen Lin. A Practical Guide to Support Vector Classification, November 2003.

[38] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. *Automated Machine Learning: Methods, Systems, Challenges*. Springer, 2018. In press, available at `http://automl.org/book`.

[39] Frank Hutter, Joaquin Vanschoren, Marius Lindauer, Charles Weill, Katharina Eggensperger, and Matthias Feurer. Icml workshop on automated machine learning. AutoML Workshop at ICML 2020, 2020.

[40] Charles Isbell. You Can't Escape Hyperparameters and Latent Variables: Machine Learning as a Software Engineering Enterprise. NeurIPS Keynote, 2020.

[41] George H. John. Cross-Validated C4.5: Using Error Estimation for Automatic Parameter Selection. Technical report, Stanford University, Stanford, CA, USA, 1994.

[42] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 12 2014.

[43] Barteld P. Koou. Probabilistic Dynamic Epistemic Logic. *Journal of Logic Language and Information*, 12(4):381–408, 2003.

[44] Leslie Lamport. "sometime" is Sometimes "Not Never": On the Temporal Logic of Programs. In *Proceedings of the 7th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '80, page 174–185, New York, NY, USA, 1980. Association for Computing Machinery. ISBN 0897910117. doi: 10.1145/567446.567463.

[45] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An Empirical Evaluation of Deep Architectures on Problems with Many factors of Variation. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 473–480, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933.

[46] Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient BackProp. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, page 9–50, Berlin, Heidelberg, 1998. Springer-Verlag. ISBN 3540653112.

[47] Keith Lehrer. The Gettier Problem and the Analysis of Knowledge. In George Sotiros Pappas, editor, *Justification and Knowledge: New Studies in Epistemology*, pages 65–78. Springer Netherlands, Dordrecht, 1979.

[48] Zachary C. Lipton and Jacob Steinhardt. Troubling Trends in Machine Learning Scholarship. *ACM Queue*, 2018.

[49] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs Created Equal? A Large-Scale Study. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 700–709. Curran Associates, Inc., 2018.

[50] Gábor Melis, Chris Dyer, and Phil Blunsom. On the State of the At of Evaluation in Neural Language models. In *International Conference on Learning Representations*, 2018.

[51] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[52] Tom M. Mitchell. The Need for Biases in Learning Generalizations. Technical report, Rutgers University, New Brunswick, NJ, 1980. `http://www-cgi.cs.cmu.edu/~tom/pubs/NeedForBias_1980.pdf`.

[53] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A Metric Learning Reality Check, 2020.

[54] Todd Mytkowicz, Amer Diwan, Matthias Hauswirth, and Peter F. Sweeney. Producing Wrong Data without Doing Anything Obviously Wrong! In *Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS XIV, page 265–276, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584065.

[55] Susan Owicki and Leslie Lamport. Proving Liveness Properties of Concurrent Programs. *ACM Trans. Program. Lang. Syst.*, 4(3):455–495, July 1982. ISSN 0164-0925. doi: 10.1145/357172. 357178.

[56] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, November 2011. ISSN 1532-4435.

[57] Joelle Pineau. The Machine Learning Reproducibility Checklist, March 2019. `https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf`.

[58] Amir Pnueli. The Temporal Logic of Programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, SFCS '77, page 46–57, USA, 1977. IEEE Computer Society.

[59] Carlos Riquelme, George Tucker, and Jasper Snoek. Deep Bayesian Bandits Showdown: An Empirical Comparison of Bayesian Deep Networks for Thompson Sampling. In *ICLR*, February 2018.

[60] F. Schneider, L. Balles, and P. Hennig. DeepOBS: A Deep Learning Optimizer Benchmark Suite. In *7th International Conference on Learning Representations (ICLR)*. ICLR, May 2019.

[61] Dana Scott. Advice on modal logic. In Karel Lambert, editor, *Philosophical Problems in Logic: Some Recent Developments*, pages 143–173. Springer Netherlands, Dordrecht, 1970.

[62] D. Sculley, Jasper Snoek, Alex Wiltschko, and Ali Rahimi. Winner's Curse? On Pace, Progress, and Empirical Rigor. In *ICLR 2018 Workshop*, February 2018.

[63] Krister Segerberg. Two Traditions in the Logic of Belief: Bringing them Together. In Hans Jürgen Ohlbach and Uwe Reyle, editors, *Logic, Language and Reasoning: Essay in Honour of Dov Gabbay*, pages 135–148. Springer Science+Business Media, Dordrecht, Netherlands, 1999.

[64] K Sinha, J Pineau, J Forde, R N Ke, and H Larochelle. NeurIPS 2019 Reproducibility Challenge. *ReScience*, 6(2), 2020.

[65] Prabhu Teja Sivaprasad, Florian Mai, Thijs Vogels, Martin Jaggi, and François Fleuret. Optimizer Benchmarking Needs to Account for Hyperparameter Tuning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9036–9045. PMLR, 13–18 Jul 2020.

[66] Brian Cantwell Smith. The Limits of Correctness. *SIGCAS Computer and Society*, page 18–26, January 1985. ISSN 0095-2737. doi: 10.1145/379486.379512. URL `https://doi.org/10.1145/379486.379512`.

[67] Raymond M. Smullyan. Logicians Who Reason about Themselves. In *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning about Knowledge*, TARK '86, page 341–352, San Francisco, CA, USA, 1986. Morgan Kaufmann Publishers Inc. ISBN 0934613049.

[68] Richmond H. Thomason. Combinations of tense and modality. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic: Volume II: Extensions of Classical Logic*, pages 135–165. Springer Netherlands, Dordrecht, 1984. ISBN 978-94-009-6259-0.

[69] Johan van Benthem. Epistemic Logic and Epistemology: The State of Their Affairs. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition*, 128(1):49–76, 2006.

[70] Johan van Benthem. Dynamic logic for belief revision. *Journal of Applied Non-Classical Logics*, 17(2):129–155, 2007.

[71] Johan van Benthem, Jelle Gerbrandy, and Barteld P. Kooi. Dynamic Update with Probabilities. *Stud Logica*, 93(1):67–96, 2009.

[72] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The Marginal Value of Adaptive Gradient Methods in Machine Learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4148–4158. Curran Associates, Inc., 2017.

[73] Cristobal Young. Model Uncertainty and the Crisis in Science. *Socius*, 4:2378023117737206, 2018. doi: 10.1177/2378023117737206.

[74] Zhao Zhong, Junjie Yan, Wei Wu, Jing Shao, and Cheng-Lin Liu. Practical block-wise neural network architecture generation. *CVPR*, 2018.

[75] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017.

Checklist

1. For all authors...

    (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

    (b) Did you describe the limitations of your work? [Yes] See Section 5.

    (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6. We also expand further in the Appendix.

    (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

    (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sections 3, 4 & 5. For extended theoretical results and assumptions, see the Appendix.

    (b) Did you include complete proofs of all theoretical results? [Yes] See Section 5 and the Appendix.

3. If you ran experiments...

    (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide detailed information concerning the code and data in the Appendix. We release the code at https://github.com/pasta41/deception and provide further instructions in the README.

    (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the Appendix.

    (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 2 and the Appendix.

    (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

    (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 2 and the Appendix.

    (b) Did you mention the license of the assets? [Yes] See Appendix.

    (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We provide detailed information concerning the code and data in the Appendix. We release the code at https://github.com/pasta41/deception and provide further instructions in the README.

    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We used benchmark datasets.

    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Glossary

We provide a glossary of terms, definitions, and symbols that we introduce throughout the paper, streamlined in one place.

A.1 Definitions Reference

hyper-hyperparameters (hyper-HPs) are HPO-procedure-input values, such as the spacing between different points in the grid for grid search and the distributions to sample from in random search.

Definition 1. *A log ℓ records all the choices and measurements made during an HPO run, including the total time T it took to run. It has all necessary information to make the HPO run reproducible.*

Definition 2. *An HPO procedure H is a tuple $(H_*, \mathcal{C}, \Lambda, \mathcal{A}, \mathcal{M}, G, X)$ where H_* is a randomized algorithm, $\mathcal{C}$ is a set of allowable hyper-HPs (i.e., allowable configurations for H_*), Λ is a set of allowable HPs (i.e., of HP sets λ), $\mathcal{A}$ is a training algorithm (e.g. SGD), $\mathcal{M}$ is a model (e.g. VGG16), G is a PRNG, and X is some dataset (usually split into train and validation sets). When run, H_* takes as input a hyper-HP configuration $c \in \mathcal{C}$ and a random seed $r \in \mathcal{I}$, then proceeds to run $\mathcal{A}_\lambda$ (on $\mathcal{M}_\lambda$ using $G(r)$ and data[9] from X) some number of times for different HPs $\lambda \in \Lambda$. Finally, H_* outputs a tuple (λ^*, ℓ), where λ^* is the HP configuration chosen by HPO and ℓ is the log documenting the run.*

Definition 3. *An **epistemic hyperparameter optimization procedure (EHPO)** is a tuple $(\mathcal{H}, \mathcal{F})$ where $\mathcal{H}$ is a set of HPO procedures H (Definition 2) and $\mathcal{F}$ is a function that maps a set of HPO logs $\mathcal{L}$ (Definition 1) to a set of logical formulas $\mathcal{P}$, i.e. $\mathcal{F}(\mathcal{L}) = \mathcal{P}$. An execution of EHPO involves running each $H \in \mathcal{H}$ some number of times (each run produces a log ℓ) and then evaluating $\mathcal{F}$ on the set of logs $\mathcal{L}$ produced in order to output the conclusions $\mathcal{F}(\mathcal{L})$ we draw from all of the HPO runs.*

Definition 4. *A randomized **strategy σ** is a function that specifies which action the demon will take. Given $\mathcal{L}$, its current set of logs, $\sigma(\mathcal{L})$ gives a distribution over concrete actions, where each action is either 1) running a new H with its choice of hyper-HPs c and seed r 2) erasing some logs, or 3) returning. We let Σ denote the set of all such strategies.*

Definition 5. *Let $\sigma[\mathcal{L}]$ denote the logs output from executing strategy σ on logs $\mathcal{L}$, and let $\tau_\sigma(\mathcal{L})$ denote the total time spent during execution. $\tau_\sigma(\mathcal{L})$ is equivalent to the sum of the times T it took each HPO procedure $H \in \mathcal{H}$ executed in strategy σ to run. Note that both $\sigma[\mathcal{L}]$ and $\tau_\sigma(\mathcal{L})$ are random variables, as a function of the randomness of selecting G and the actions sampled from $\sigma(\mathcal{L})$. For any formula p and any $t \in \mathbb{R}_{>0}$, we say $\mathcal{L} \models \Diamond_t p$, i.e. "$\mathcal{L}$ models that it is possible p in time t," if*

$$\text{there exists a strategy } \sigma \in \Sigma, \text{ such that } \mathbb{P}(\sigma[\mathcal{L}] \models p) = 1 \text{ and } \mathbb{E}[\tau_\sigma(\mathcal{L})] \leq t.$$

Definition 6. *For any formula p, we say $\mathcal{L} \models \mathcal{B}p$, "$\mathcal{L}$ models our belief in p", if $p \in \mathcal{F}(\mathcal{L})$.*

Definition 7. *Suppose that we are given a naive EHPO procedure $(\{H\}, \mathcal{F}_n)$, in which H is random search and is the only HPO in our EHPO, and $\mathcal{F}_n$ is a "naive" belief function associated with a naive reasoner $\mathcal{B}_n$. For any $K, R \in \mathbb{N}$, we define the "(K, R)-defended" belief function $\mathcal{F}_*$ for a skeptical reasoner $\mathcal{B}_*$ as the following conclusion-drawing procedure. First, $\mathcal{F}_*$ only makes conclusion set $\mathcal{P}_*$ from a single log $\hat{\ell}$ with $K * R$ trials; otherwise, it concludes nothing, outputting $\emptyset$. Second, $\mathcal{F}_*$ splits the single $\hat{\ell}$ into R logs $\ell_1, \ell_2, \ldots, \ell_R$, each containing K independent-random-search trials.[10] Finally, $\mathcal{F}_*$ outputs the intersection of what the naive reasoner would have output on each log ℓ_i,*

$$\mathcal{F}_*(\{\hat{\ell}\}) = \mathcal{P}_* \equiv \mathcal{F}_n(\{\ell_1\}) \cap \mathcal{F}_n(\{\ell_2\}) \cap \cdots \cap \mathcal{F}_n(\{\ell_R\}).$$

Equivalently, $\{\hat{\ell}\} \models \mathcal{B}_* p$ only if $\{\ell_i\} \models \mathcal{B}_n p$ for all i.

---

[9]Definition 2 does not preclude cross-validation, as this can be part of H_*. The input dataset X can be split in various ways, as a function of the random seed r.

[10]This is not generally allowable. $\mathcal{F}_*$ can do this because random-search logs contain interchangeable trials.

A.2 Symbols and Acronyms Reference

Term	Explanation	Example
HPO	Acronym for hyperparameter optimization	
$\mathcal{J}, \mathcal{K}$	Used as examples of arbitrary optimizers	
P	Arbitrary atomic proposition	
p, q, ϕ	Used as arbitrary or (when specified) specific logical formulas	p = "Non-adaptive optimizers have higher test accuracy than adaptive optimizers."
HP(s)	Acronym for hyperparameter(s)	
$\ell \in \mathcal{L}$,	Log (Definition 1); log set	Figure 5; Log for running HPO using SGD
T	The total time it took to run HPO to produce a log ℓ	
$\mathcal{I}$	A set of integers	Typically the 64-bit integers
r	Random seed; $r \in \mathcal{I}$	
G	Pseudo-random number generator; $G(r)$; $G : \mathcal{I} \to \mathcal{I}^\infty$	
PRNG	Acronym for pseudo-random number generator; G	
H	HPO procedure (Definition 2)	SGD, VGG-16 grid search experiment
H_*	A randomized algorithm used in H	Random search
$c \in \mathcal{C}$	Hyper-HP configuration; of set of allowable such configurations for H_*	powers-of-2 grid spacing; configurations the demon has access to
$\lambda \in \Lambda; \lambda^*$	HP config. used to run an HPO pass; of allowable HP configs., determined by c; λ^* is the output HP config. that performs the best	$\alpha = 1$ in Wilson; allowable α values, e.g. $[.001, .01, 1]$
$\mathcal{A}; \mathcal{A}_\lambda$	Training algorithm; parameterized by HPs λ	SGD; SGD with $\alpha = 1$
$\mathcal{M}; \mathcal{M}_\lambda$	Model; parameterized by HPs λ	VGG16
X	A dataset	CIFAR-10
α	Learning rate	Figure 2, $\alpha = 1$
ϵ	Adam-specific HP	Figure 2, we set $\epsilon = 10^{12}$
EHPO	Epistemic HPO (Definition 3)	Our defended random search in Section 5
$\mathcal{H}$	Set of HPO procedures H	
$\mathcal{P}$	Set of concluded logical formulas; $p \in \mathcal{P}$	
$\mathcal{F}$	A function that maps a set of HPO logs $\mathcal{L}$ to a set of logical formulas $\mathcal{P}$	$\mathcal{F}_*$ (skeptical belief function); $\mathcal{F}_n$ (naive belief function)
$\square$	Modal logic operator for "necessary"	$\square p$ reads "It is necessary that p
$\Diamond$	Modal logic operator for "possible"	$\Diamond p$ reads "It is possible that p
$\vdash$	Indicates a theorem of propositional logic	$\vdash Q \to \square Q$ (necessitation)
$\Diamond_t$	EHPO modal operator (Section 4.2; Definition 5)	
$\mathcal{B}$	Belief modal operator	$\mathcal{B}_*$ (skeptical belief); $\mathcal{B}_n$ (naive belief)
$\sigma \in \Sigma$	A randomized strategy function that specifies EHPO actions; set of all such strategies (Section 4.2, Definition 4)	
$\sigma(\mathcal{L})$	Distribution over concrete actions for log set	
$\sigma[\mathcal{L}]$	The logs output from running σ on $\mathcal{L}$	
$\tau_\sigma(\mathcal{L})$	Total time spent executing $\sigma[\mathcal{L}]$	
$\models$	Denotes "models"	$\mathcal{L} \models \Diamond_t p$: $\mathcal{L}$ model that p is possible in t
γ	Renyi-∞-divergence constant upper bound (Theorem 1)	
K, R	Numbers of independent random search trials (Section 5)	
κ	Subsampling size (Algorithm 1)	We set $\kappa = 11$ (Section 5)
M	Subsampling budget (Algorithm 1)	We set $M = 10000$ (Section 5)
δ	Skeptical reasoner conclusion threshold (Algorithm 1)	See Table 1

B Section 2 Appendix: Additional Notes on the Preliminaries

The code for running these experiments can be found at `https://github.com/pasta41/deception`.

B.1 Empirical Deception Illustration using Wilson et al. [72]

B.1.1 Why we chose Wilson et al. [72]

We elaborate on why we specifically chose Wilson et al. [72] as our running example of hyperparameter deception. There are four main reasons why we thought this was the right example to focus on for an illustration. First, the experiment involves optimizers known across ML (e.g. SGD, Adam), a model frequently used for benchmark tasks (VGG16) and a commonly-used benchmark dataset (CIFAR-10). Unlike other examples of hyperparameter deception, one does not need highly-specialized domain knowledge to understand the issue [16, 49]. Second, the paper is exceptionally well-cited and known in the literature, so many folks in the community are familiar with its results. Third, we were certain that we could demonstrate hyperparameter deception before we ran our experiments; we observe that Adam's update rule basically simulates Heavy Ball when its ϵ parameter is set high enough. So, we were confident that we could (at the very least) get Adam to perform as well as Heavy Ball via changing hyper-HPs, which would demonstrate hyperparameter deception. We then found further support for this observation in concurrent work [65], which cited earlier work [11] that also observes this. Fourth, the claim in Wilson et al. [72] is fairly broad. They make a claim about adaptive vs. non-adaptive optimizers, more generally. If the claim had been narrower – about small ϵ values for numerical stability, then perhaps hyperparameter deception would not have occurred. In general, we note that narrower claims could help avoid deception.

B.2 Expanded empirical results

We elaborate on the results we present in Section 2.

B.2.1 Experimental setup

We replicate and run a variant of Wilson et al. [72]'s VGG16 experiment on CIFAR-10, using SGD, Heavy Ball, and Adam as the optimizers.

We launch each run on a local machine configured with a 4-core 2.6GHz Inter (R) Xeon(R) CPU, 8GB memory and an NIVIDIA GTX 2080Ti GPU.

Following the exact configuration from Wilson et al. [72], we set the mini-batch size to be 128, the momentum constant to be 0.9 and the weight decay to be 0.0005.

The learning rate is scheduled to follow a linear rule: The learning rate is decayed by a factor of 10 every 25 epochs. The total number of epochs is set to be 250.

For the CIFAR-10 dataset, we apply random horizontal flipping and normalization. Note that Wilson et al. [72] does not apply random cropping on CIFAR-10; thus we omit this step to be consistent with their approach. We adopt the standard cross entropy loss.

For each HPO setting, we run 5 times and average the results and include error bars two standard deviations above and below the mean.

B.2.2 Associated results and logs

In line with our notion of a *log* (Definition 1), we provide data tables (Figures 5, 6, and 7) that correspond with our results graphed in the Figures 2, 3, 4.
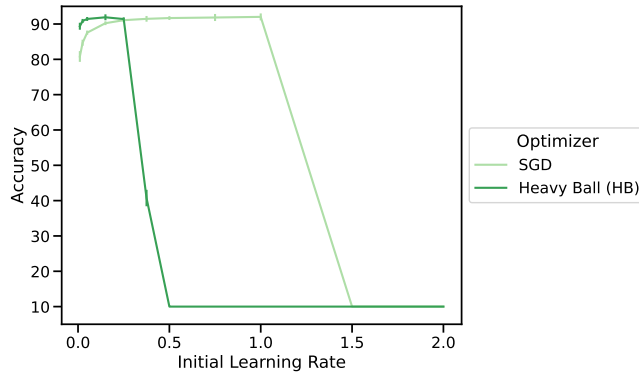
Figure 2: Full test accuracy results of VGG-16 on CIFAR-10 for SGD and Heavy Ball learning rate (α) HPO. Error bars indicate two standard deviations above and below the mean. Each HPO setting is measured with five replicates. We achieve similar performance as Wilson et al. [72].
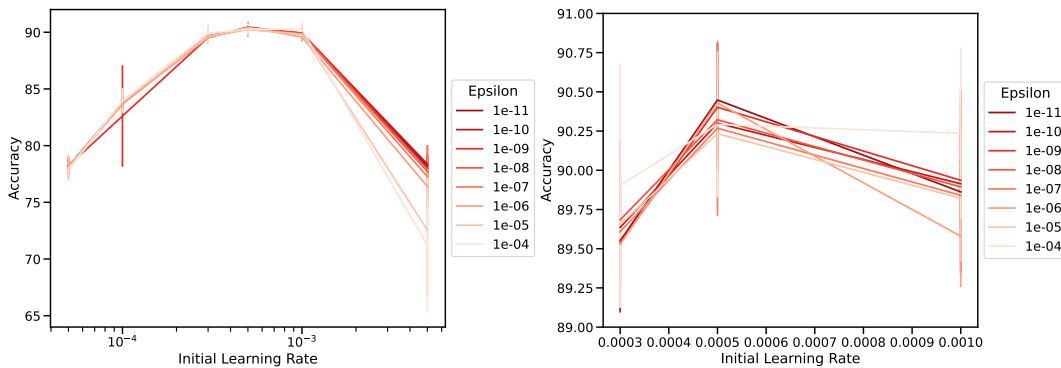
Figure 3: Tuning over learning rate for different small values of ϵ. On the left, we show a wide range of learning rates tested. On the right, we zoom in on the portion of results where the best test accuracy occurs. These results reflect what Wilson et al. [72] showed, but with tuning over ϵ (small values). Each HP setting is used to train VGG-16 on CIFAR-10 five times, and the error bars represent two standard deviations above and below the mean test accuracy.
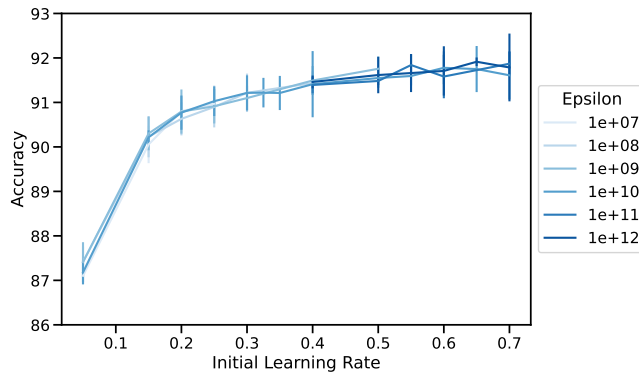
Figure 4: Results for our expanded search over large ϵ values for Adam. We show test accuracy on CIFAR-10 as a function of different learning rates α for the different large ϵ values. Error bars show two standard deviations above and below mean test accuracy for five replicates for each HP setting.

20
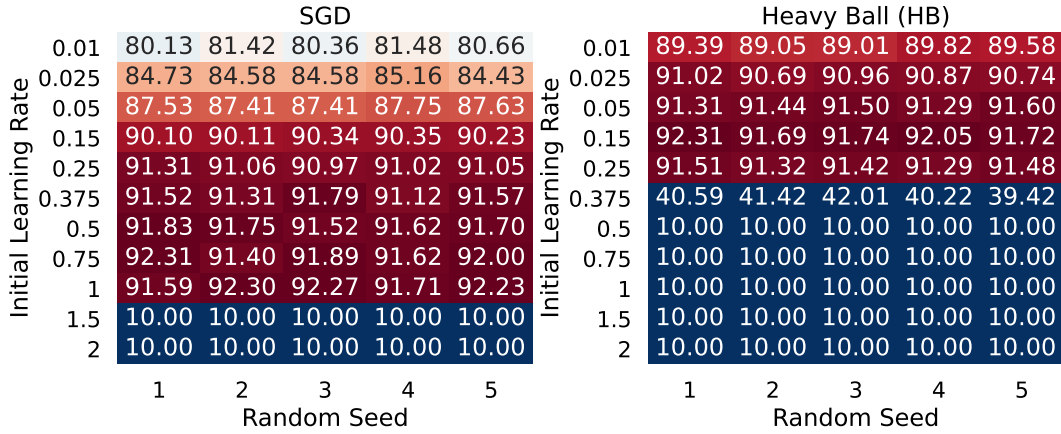
Figure 5: Heatmap logs of test accuracy of VGG-16 on CIFAR-10 for SGD and Heavy Ball for each initial learning rate and random seed. These logs correspond to the results graphed in Figure 2.

Figure 6: Heatmap logs of test accuracy of VGG-16 on CIFAR-10 for Adam for each initial learning rate and random seed for different small values of Adam's ϵ HP. These results correspond to those graphed in Figure 3.

epsilon = 1e-04

Initial Learning Rate	1	2	3	4	5
5e-05	78.04	77.53	77.66	78.16	77.81
1e-04	83.58	83.87	83.62	84.85	83.89
3e-04	90.16	89.65	89.97	89.39	90.35
5e-04	90.23	90.09	90.22	90.23	90.69
1e-03	90.71	90.03	90.16	90.11	90.17
5e-03	73.25	75.23	70.16	69.55	67.94

epsilon = 1e-05

Initial Learning Rate	1	2	3	4	5
5e-05	78.99	78.03	77.64	77.76	77.91
1e-04	83.80	83.85	83.65	84.15	83.80
3e-04	89.55	90.08	89.45	89.73	89.46
5e-04	90.41	90.09	90.22	90.28	90.17
1e-03	89.60	89.68	89.85	89.86	90.12
5e-03	70.79	76.68	69.19	73.55	72.53

epsilon = 1e+07

Initial Learning Rate	1	2	3	4	5
5e-02	87.17	87.16	87.02	87.15	87.01
1e-01	89.96	89.97	89.87	90.10	90.38
2e-01	90.68	90.94	90.62	90.82	90.95
2e-01	90.90	91.17	90.66	91.03	91.08

epsilon = 1e+08

Initial Learning Rate	1	2	3	4	5
5e-02	87.32	87.55	87.20	87.57	87.44
1e-01	90.15	90.32	89.96	90.55	90.17
2e-01	90.75	90.45	90.43	90.76	90.76
2e-01	90.94	90.86	90.92	91.19	90.58
3e-01	91.05	91.01	91.28	91.52	91.22
4e-01	91.50	91.50	91.47	91.48	91.22

epsilon = 1e+09

Initial Learning Rate	1	2	3	4	5
5e-02	87.08	87.38	87.54	87.41	87.64
1e-01	90.50	90.43	90.31	90.05	90.22
2e-01	90.94	90.86	90.77	90.40	91.00
2e-01	90.83	91.16	90.68	90.89	91.01
3e-01	91.03	91.22	90.91	91.16	91.17
4e-01	91.63	91.41	91.54	91.27	91.61
5e-01	91.74	91.72	91.86	91.89	91.57

epsilon = 1e+10

Initial Learning Rate	1	2	3	4	5
5e-02	87.30	87.25	87.20	86.98	87.16
	90.19	90.24	90.30	90.23	90.12
2e-01	90.55	91.01	90.77	90.87	90.65
	91.04	91.18	90.87	91.18	90.87
3e-01	91.24	91.44	91.33	90.98	91.08
	91.05	91.40	91.35	91.21	91.09
3e-01	91.13	91.27	91.11	91.50	91.05
	91.94	91.11	91.08	91.59	91.33
5e-01	91.56	91.40	91.53	91.58	91.67
	91.58	91.45	91.63	91.60	91.71
6e-01	92.02	91.79	91.83	91.70	91.55
	91.94	91.91	91.70	91.86	91.34
7e-01	91.59	91.41	92.06	91.53	91.46

epsilon = 1e+11

Initial Learning Rate	1	2	3	4	5
4e-01	91.42	91.46	91.46	91.27	91.35
5e-01	91.46	91.50	91.48	91.53	91.44
6e-01	91.66	91.93	91.92	91.88	91.81
6e-01	91.49	91.72	91.47	91.91	91.32
7e-01	91.73	91.98	92.01	91.80	91.83

epsilon = 1e+12

Initial Learning Rate	1	2	3	4	5
4e-01	91.47	91.49	91.46	91.52	91.36
5e-01	91.43	91.69	91.58	91.91	91.47
6e-01	91.46	91.48	91.89	91.85	91.62
6e-01	91.35	92.00	91.82	91.85	91.52
7e-01	91.95	91.88	91.94	91.93	91.87
7e-01	91.88	91.20	91.73	91.92	92.20

Random Seed

Figure 7: Heatmap logs of test accuracy of VGG-16 on CIFAR-10 for Adam for each initial learning rate and random seed for different values of Adam's ϵ using our expanded search space. These logs reflect the results graphed in Figure 4.

B.3 Empirical Deception Illustration using Merity et al. [51]

In addition to the computer vision experiments of Wilson et al. [72], we also show a separate line of experiments from NLP: training an LSTM on Wikitext-2 using Nesterov and Heavy Ball as the optimizers. We illustrate deception (i.e., the possibility of drawing inconsistent conclusions) using two different sets of hyper-HPs to configure HPO grids for tuning the learning rate. We run ten replicates for each optimizer / grid combination (a total of 40 runs). We run these experiments using the same hardware as described in Appendix B.2.1.
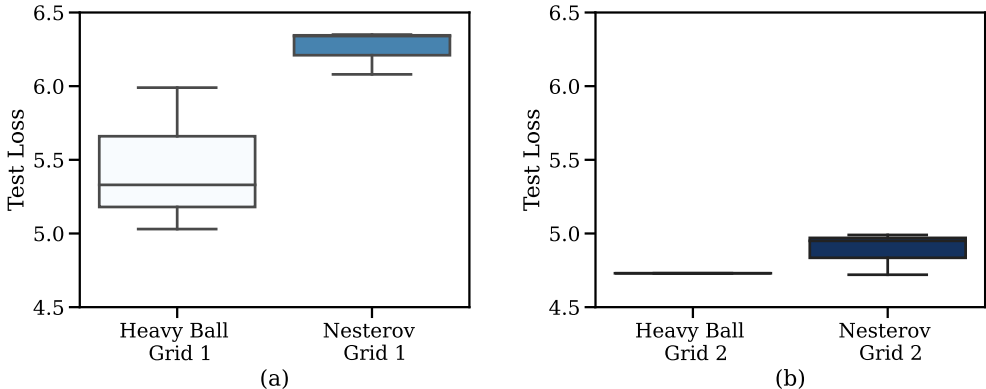
Figure 8: Demonstrating the possibility of drawing inconsistent conclusions from HPO (what we shorthand *hyperparameter deception*) LSTM on Wikitext-2 using Nesterov and Heavy Ball as the optimizers. Each box plot represents a log. In (a), we use the grid $\alpha = 1, 5, 10, 15, 20, 25, 30, 35, 40$, from which we can reasonably conclude that Nesterov outperforms HB. In (b), we use the grid $\alpha = 10, 20, 30, 40$, from which we can reasonably conclude that HB outperforms Nesterov.

C Section 3 Appendix: Epistemic Hyperparameter Optimization

C.1 Additional concrete interpretations of EHPO

For concision, in the main text we focus on examples of EHPO procedures that compare the performance of different optimizers. However, it is worth noting that our definition of EHPO (Definition 3) is more expansive than this setting. For example, it is possible to run EHPO to compare different models (perhaps, though not necessarily, keeping the optimizer fixed), to draw conclusions about the relative performance of different models on different learning tasks.

C.2 Descartes' Evil Demon Thought Experiment

Our formalization was inspired by Descartes' evil genius/demon thought experiment. This experiment more generally relates to his use of systematic doubt in *The Meditations* more broadly. It is this doubt/skepticism (and its relationship to possibility) that we find useful for the framing of an imaginary, worst-case adversary. In particular, we draw on the following quote, from which we came up with the term *hyperparameter deception*:

> *I will suppose...an evil genius, supremely powerful and clever, who has directed his entire effort at deceiving me. I will regard the heavens, the air, the earth, colors, shapes, sounds, and all external things as nothing but the bedeviling hoaxes of my dreams, with which he lays snares for my credulity...even if it is not within my power to know anything true, it certainly is within my power to take care resolutely to withhold my assent to what is false, lest this deceiver, however powerful, however clever he may be, have any effect on me.* – Descartes

Figure 9: How the authors imagine the EHPO-running demon

For more on the long (and rich) history of the use of imaginary demons and devils as adversaries—notably a different conception of an adversary than the potential real threats posed in computer security research—we refer the reader to Canales [9].

D   Section 4 Appendix: Modal Logic Formalization

D.1   Further Background on Modal Logic

We first provide the necessary background on modal logic, which will inform the proofs in this appendix (Appendix D.1.1). We then describe our possibility logic—a logic for representing the possible results of the evil demon running EHPO—and prove that it is a valid modal logic (Appendix D.2.1). We then present a primer on modal belief logic (Appendix D.2.4), and suggest a proof for the validity of combining our modal possibility logic with modal belief logic (Appendix D.2.5).

D.1.1   Axioms from Kripke Semantics

Kripke semantics in modal logic inherits all of the the axioms from propositional logic, which assigns values T and F to each atom p, and adds two operators, one for representing *necessity* ($\Box$) and one for *possibility* ($\Diamond$).

- $\Box p$ reads "It is necessary that p".
- $\Diamond p$ reads "It is possible that p".

The $\Diamond$ operator is just syntactic sugar, as it can be represented in terms of $\neg$ and $\Box$:

$$\Diamond p \equiv \neg\Box\neg p \tag{1}$$

which can be read as:

"It is possible that p" is equivalent to "It is not necessary that not p."

The complete set of rules is as follows:

- Every atom p is a sentence.
- If D is a sentence, then
  - $\neg D$ is a sentence.
  - $\Box D$ is a sentence.
  - $\Diamond D$ is a sentence.
- If D and E are sentences, then
  - $D \wedge E$ is a sentence.
  - $D \vee E$ is a sentence.

24

- $D \rightarrow E$ is a sentence.
- $D \leftrightarrow E$ is a sentence

- $\Box(\mathcal{D} \rightarrow \mathcal{E}) \rightarrow (\Box\mathcal{D} \rightarrow \Box\mathcal{E})$ (Distribution)

- $\mathcal{D} \rightarrow \Box\mathcal{D}$ (Necessitation)

D.1.2 Possible Worlds Semantics

Modal logic introduces a notion of *possible worlds*. Broadly speaking, a possible world represents the state of how the world *is* or potentially *could be* [10, 24]. Informally, $\Box D$ means that D is true at *every* world (Equation 2); $\Diamond D$ means that D is true at *some* world (Equation 3).

Possible worlds give a different semantics from more familiar propositional logic. In the latter, we assign truth values $\{T, F\}$ to propositional variables $p \in \mathcal{P}$, from which we can construct and evaluate sentences $D \in \mathcal{D}$ in a truth table. In the former, we introduce a set of possible worlds, $\mathcal{W}$, for which each $w \in \mathcal{W}$ has own truth value for each p. This means that the value of each p can differ across different worlds w. Modal logic introduces the idea of valuation function,

$$\mathcal{V} : (\mathcal{W} \times \mathcal{D}) \rightarrow \{T, F\}$$

to assign truth values to logical sentences at different worlds. This in turn allows us to express the formulas, axioms, and inference rules of propositional logic in terms of $\mathcal{V}$. For example,

$$\mathcal{V}(w, \neg D) = T \leftrightarrow \mathcal{V}(w, D) = F$$

There are other rules that each correspond to a traditional truth-table sentence evaluation, but conditioned on the world in which the evaluation occurs. We omit these for brevity and refer the reader to Chellas [10].

We do include the valuation rules for the $\Box$ and $\Diamond$ operators that modal logic introduces (Equations 2 & 3). To do so, we need to introduce one more concept: The accessibility relation, $\mathcal{R}$. $\mathcal{R}$ provides a frame of reference for one particular possible world to access other possible worlds; it is a way from moving from world to world. So, for an informal example, $\mathcal{R}w_1w_2$ means that w_2 is possible relative to w_1, i.e. we can reach w_2 from w_1. Such a relation allows for a world to be possible relative potentially to some worlds but not others. More formally,

$$R \subseteq \mathcal{W} \times \mathcal{W}$$

Overall, the important point is that we have a collection of worlds $\mathcal{W}$, an accessibility relation $\mathcal{R}$, and a valuation function $\mathcal{V}$, which together defines a Kripke model, which captures this system:

$$\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle$$

Finally, we can give the valuation function rules for $\Box$ and $\Diamond$:

$$\mathcal{V}(w, \Box D) = T \leftrightarrow \forall w', (\mathcal{R}ww' \rightarrow \mathcal{V}(w', D) = T) \tag{2}$$

$$\mathcal{V}(w, \Diamond D) = T \leftrightarrow \exists w', (\mathcal{R}ww' \wedge \mathcal{V}(w', D) = T) \tag{3}$$

Informally, for $\Box D$ to be true in a world, it must be true in every possible world that is reachable by that world. For $\Diamond D$ to be true in a world, it must be true in some possible world that is reachable by that world.

D.2 Our Multimodal Logic Formulation

D.2.1 A Logic for Reasoning about the Conclusion of EHPO

As in Section 4, we can define the well-formed formulas of our indexed modal logic[11] recursively in Backus-Naur form, where t is any real number and P is any atomic proposition

$$\kappa := P \mid \neg\kappa \mid \kappa \wedge \kappa \mid \Diamond_t \kappa \tag{4}$$

where κ is a well-formed formula.

As we note in Section 4, where we first present this form of defining modal-logic, $\square$ is syntactic sugar, with $\square p \equiv \neg\Diamond\neg p$ (which remains true for our indexed modal logic). Similarly, "or" has $p \vee q \equiv \neg(\neg p \wedge \neg q)$ and "implies" has $p \rightarrow q \equiv \neg p \vee q$, which is why we do not include them for brevity in this recursive definition.

We explicitly define the relevant semantics for $\Diamond_t$ for reasoning about the demon's behavior in running EHPO. For clarity, we replicate that definition of the semantics of expressing the possible outcomes of EHPO conducted in bounded time (Definitions 4 & 5, respectively) below:

Definition. *A randomized **strategy** σ is a function that specifies which action the demon will take. Given $\mathcal{L}$, its current set of logs, $\sigma(\mathcal{L})$ gives a distribution over concrete actions, where each action is either 1) running a new H with its choice of hyper-HPs c and seed r 2) erasing some logs, or 3) returning. We let Σ denote the set of all such strategies.*

We can now define what the demon can reliably bring about, in terms of executing a strategy in bounded time:

Definition. *Let $\sigma[\mathcal{L}]$ denote the logs output from executing strategy σ on logs $\mathcal{L}$, and let $\tau_\sigma(\mathcal{L})$ denote the total time spent during execution. $\tau_\sigma(\mathcal{L})$ is equivalent to the sum of the times T it took each HPO procedure $H \in \mathcal{H}$ executed in strategy σ to run. Note that both $\sigma[\mathcal{L}]$ and $\tau_\sigma(\mathcal{L})$ are random variables, as a function of the randomness of selecting G and the actions sampled from $\sigma(\mathcal{L})$. For any formula p and any $t \in \mathbb{R}_{>0}$, we say $\mathcal{L} \models \Diamond_t p$, i.e. "$\mathcal{L}$ models that it is possible p in time t," if*

$$\text{there exists a strategy } \sigma \in \Sigma, \text{ such that } \quad \mathbb{P}(\sigma[\mathcal{L}] \models p) = 1 \text{ and } \mathbb{E}[\tau_\sigma(\mathcal{L})] \leq t.$$

D.2.2 A Possible Worlds Interpretation

Drawing on the possible worlds semantics that modal logic provides (Section D.1.2), we can define specific possible worlds semantics for our logic for expressing the actions of the demon in EHPO from above.

Definition 8. *A **possible world** represents the set of logs $\mathcal{L}$ the demon has produced at time $\tau_\sigma(\mathcal{L})$ (i.e., after concluding running EHPO) and the set of formulas $\mathcal{P}$ that are modeled from $\mathcal{L}$ via $\mathcal{F}$.*

Therefore, different possible worlds represent the states that *could have existed* if the evil demon had executed different strategies (Definition 4). In other words, if it had performed EHPO with different learning algorithms, different HPO procedures, different hyper-hyperparameter settings, different amounts of time (less than the total upper bound), different learning tasks, different models, etc... to produce a different set of logs $\mathcal{L}$ and corresponding set of conclusions $\mathcal{P}$.

In this formulation, the demon has knowledge of all possible worlds; it is trying to fool us about the relative performance of algorithms by showing as an intentionally deceptive world. Informally, moving from world to world (via an accessibility relation) corresponds to the demon running more passes of HPO to produce more logs to include in $\mathcal{L}$.

D.2.3 Syntax and Semantics for the Logic Modeling the Demon Running the EHPO

We provide proofs and intuitions of the axioms of our EHPO logic in this section, based on a correspondence with un-indexed modal logic.

---

[11]For an example of another indexed modal logic concerning probability, please refer to Heifeitz and Mongin [33].

We remind the reader that the following are the axioms of our indexed modal logic:

$$\vdash (p \to q) \to (\Diamond_t p \to \Diamond_t q) \qquad \textit{(necessitation + distribution)}$$
$$p \to \Diamond_t p \qquad \textit{(reflexivity)}$$
$$\Diamond_t \Diamond_s p \to \Diamond_{t+s} p \qquad \textit{(transitivity)}$$
$$\Diamond_s \Box_t p \to \Box_t p \qquad \textit{(symmetry)},$$
$$\Diamond_t (p \land q) \to (\Diamond_t p \land \Diamond_t q) \qquad \textit{(distribution over } \land \textit{)}$$

In short, to summarize these semantics—the demon has knowledge of all possible hyper-hyperparameters, and it can pick whichever ones it wants to run EHPO within a bounded time budget t to realize the outcomes it wants: $\Diamond_t p$ means it can realize p.

We inherit distribution and necessitation from un-indexed modal logic; they are axiomatic based on Kripke semantics. We provide greater intuition and proofs below.

Notes on necessitation for $\Box_t$:

Necessitation for our indexed necessary operator can be written as follows:

$$\vdash p \to \Box_i p$$

As we note in Section 4, $\vdash$ just means here that p is a theorem of propositional logic. So, if p is a theorem, then so is $\Box_t p$. By theorem we just mean that p is provable by our axioms (these being the only assumptions we can use); so whenever p fits this definition, we can say $\Box_t p$.

For our semantics, this just means that when p is a theorem, it is necessary that p in time t.

Distribution for $\Box_t$:

$$\Box_t(p \to q) \to (\Box_t p \to \Box_t q)$$

We provide three ways to verify distribution over implication for $\Box_t$. From this, we will prove distribution over implication for $\Diamond_t$

A. The first follows from an argument about the semantics of possible worlds from the Kripke model of our system (Sections D.1.2 & D.2.2).

   i. It is fair to reason that distribution is self-evident given the definitions of implication ($\to$, formed from $\neg$ and $\lor$ in our syntax for well-formed formulas for our EPHO logic, given at (4) and necessity ($\Box_t$, formed from $\neg$ and $\Diamond_t$ in our syntax for well-formed formulas for our EHPO logic, given at (4)).

   ii. Similarly, we can further support this via our semantics of possible worlds. We can understand $\Box_t p$ to mean, informally, that it an adversary does adopt a strategy σ that is guaranteed to cause the desired conclusion p to be the case while take at most time t in expectation. Formally, as an "necessary" analog to the semantics of $\Diamond_t$ given in Definition 5:
   For any formula p, we say $\mathcal{L} \models \Box_t p$ if and only if
   $$\forall \sigma \in \Sigma, \ \mathbb{P}(\sigma[\mathcal{L}] \models p) = 1 \ \land \ \mathbb{E}[\tau_\sigma(\mathcal{L})] \leq t.$$

   Given $p \to q$ is true in **all accessible worlds** (i.e, the definition of necessary), then we can say that q is true in all accessible worlds whenever p is true in all accessible worlds. As in i. above, this just follows / is axiomatic from the definitions of necessity and implication for Kripke semantics.

B. We can also prove distribution by contradiction.

   i. Suppose that the distribution axiom does not hold. That is, the hypothesis
   $$\Box_t(p \to q)$$
   is true and the conclusion
   $$\Box_t p \to \Box_t q$$
   is false.

ii. By similar reasoning, from above $\Box_t p \to \Box_t q$ being false, we can say that $\Box_t p$ is true and $\Box_t q$ is false.

iii. We can use Modal Axiom M (reflexivity, proven in the next section) to say $\Box_t p \to p$. Since $\Box_t p$ is true, we can use *modus ponens* to determine that p is true.

iv. We can also say

$$\Box_t(p \to q) \to (p \to q) \qquad \textit{(By Modal Axiom M (reflexivity))}$$

v. Since we $\Box_t(p \to q)$ is true from above, we can conclude via *modus ponens* that $p \to q$ must also be true.

vi. We concluded above that p is true, so we can again use *modus ponens* and the fact that $p \to q$ is true to conclude that q is true.

vii. By necessitation, we can then also say $q \to \Box_t q$, and conclude that $\Box_t q$ is true. This is a contradiction, as above we said that $\Box_t$ is false.

viii. Therefore, by contradiction, $\Box_t(p \to q) \to (\Box_t p \to \Box_t q)$ is proved.

C. We can separately take an intuitionistic approach to verify the distribution axiom [1, 5]:

i. Let b be an **actual proof** of $p \to q$ so that we can say $a.b$ is a proof of $\Box_t(p \to q)$.

ii. Let d be an **actual proof** of p so that we can say $c.d$ is a proof of $\Box_t p$.

iii. From i. and ii., $b(d)$ is an **actual proof** of q, i.e. b (an actual proof of $p \to q$) is supplied d (an actual proof of p), and therefore can conclude q via an actual proof.

iv. From iii., we can say this results in a proof of $\Box_t q$, i.e. $e.[b(d)]$.

v. The above i.-iv. describes a function, $f : a.b \to f_{(a.b)}$. In other words, given **any proof** $a.b$ (i.e., of $\Box_t(p \to q)$) we can return function $f_{(a.b)}$, which turns **any proof** $c.d$ (i.e., of $\Box_t p$) into a proof $e.[b(d)]$ (i.e., of $\Box_t q$).

vi. $f_{(a.b)}$ is thus a proof of $\Box_t p \to \Box_t q$.

vii. From i.-vi., we gone from $a.b$ (a proof of $\Box_t(p \to q)$) to a proof of $\Box_t p \to \Box_t q$, i.e. have intuitionistically shown that $\Box_t(p \to q) \to (\Box_t p \to \Box_t q)$

Distribution and $\Diamond_t$:

We provide the following axiom in our logic:

$$\vdash (p \to q) \to (\Diamond_t p \to \Diamond_t q) \qquad \textit{(necessitation and distribution)}$$

and we now demonstrate it to be valid.

$$
\begin{aligned}
\vdash (p \to q) &\to \Box_t(p \to q) && \textit{(necessitation)} \\
&\to \Box_t(\neg q \to \neg p) && \textit{(modus tollens)} \\
&\to (\Box_t \neg q \to \Box_t \neg p) && \textit{(distribution)} \\
&\to (\neg \Box_t \neg p \to \neg \Box_t \neg q) && \textit{(modus tollens)} \\
&\to (\Diamond_t p \to \Diamond_t q) && (\Diamond_t a \equiv \neg \Box_t \neg a)
\end{aligned}
$$

This concludes our proof, for how the axioms are jointly stated.

Further, we could also say

$$(p \to q) \to \Diamond_t(p \to q) \qquad \textit{(Modal axiom M (reflexivity))}$$

And therefore also derive distribution over implication for possibility:

$$\Diamond_t(p \to q) \to (\Diamond_t p \to \Diamond_t q)$$

Modal Axiom M: Reflexivity

$$p \to \Diamond_t p$$

This axiom follows from how we have defined the semantics of our indexed modal logic (Definition 5). It follows from the fact that the demon could choose to do nothing.

We can provide a bit more color to the above as follows:

We can also derive this rule from necessitation, defined above (and from the general intuition / semantics of modal logic that necessity implies possibility). First, we can say that necessity implies possibility. We can see this a) from a possible worlds perspective and b) directly from our axioms. From a possible worlds perspective, this follows from the definition of the operators. Necessity means that there is truth at every accessible possible world, while possibility means there is truth at some accessible possible world, which puts that possible truth in time t as a subset of necessary truth in time t. From the axioms, we verify

$$
\begin{array}{ll}
\Box_t p \to \Diamond_t p & \text{(\textit{Theorem to verify, which also corresponds to Modal Axiom D (serial)})} \\
\neg\Box_t p \vee \Diamond_t p & \text{(\textit{Applying } p \to q \text{ \textit{is equiv.} } \neg p \vee q)} \\
\Diamond_t \neg p \vee \Diamond_t p & \text{(\textit{By modal conversion,} } \neg\Box_t p \to \Diamond_t \neg p) \\
& \text{(\textit{Which for our semantics is tautological})}
\end{array}
$$

That is, in time t it is possible that p or it is possible that p, which allows for us also to not conclude anything (in the case that the demon chooses to do nothing).

We can then say,

$$
\begin{array}{ll}
(\Box_t p \to p) \to \Diamond_t p & \text{(\textit{By necessitation and} } \Box_t p \to \Diamond_t p) \\
p \to \Diamond_t p & \text{(\textit{By concluding} } p \text{ \textit{from necessitation})}
\end{array}
$$

Another way to understand this axiom is again in terms of possible worlds. We can say in our system that every world is possible in relation to itself. This corresponds to the accessibility relation $\mathcal{R}ww$. As such, an equivalent way to model reflexivity is in terms of the following:

$$\Box_t p \to p$$

That is, if $\Box_t p$ holds in world w, then p also holds in world w, as is the case for $\mathcal{R}ww$. We can see this by proving $\Box_t p \to p$ by contradiction. Assuming this were false, we would need to construct a world w in which $\Box_t p$ is true and p is false. If $\Box_t p$ is true at world w, then by definition p is true at every world that w accesses. For our purposes, this holds, as $\Box_t p$ means that it is necessary for p to be the case in time t; any world that we access from this world w (i.e. by say increasing time, running more HPO) would require p to hold. Since $\mathcal{R}ww$ means that w accesses itself, that means that p must also be true at w, yielding the contradiction.

Modal Axiom 4: Transitivity

$$\Diamond_t \Diamond_s p \to \Diamond_{t+s} p \tag{5}$$

We can similarly understand transitivity to be valid intuitively from the behavior of the demon and in relation to the semantics of our possible worlds. We do an abbreviated treatment (in relation to what we say for reflexivity above) for brevity.

In terms of the demon, we note that in our semantics $\Diamond_t p$ means that it is possible for the demon to bring about conclusion p via its choices in time t. Similarly, we could say the same for $\Diamond_s p$; this means it is possible for the demon to bring about conclusion p in time s. If it is possible in time t that

it is possible in time s to bring about p, this is equivalent in our semantics to saying that it is possible in time $t + s$ to bring about conclusion p.

We can understand this rule (perhaps more clearly) in terms of possible worlds and accessibility relations.

For worlds w_n,

$$\forall w_1, \forall w_2, \forall w_3, \mathcal{R}w_1w_2 \land \mathcal{R}w_2w_3 \to \mathcal{R}w_1w_3$$

In other words, this accessibility relation indicates that if w_1 accesses w_2 and if w_2 accesses w_3, then w_1 accesses w_3.

For understanding this in terms of relative possibility, we could frame this as, if w_3 is possible relative to w_2 and if w_2 is possible relative to w_1, then w_3 is possible relative to w_1. For our semantics of the demon, this means that in some time if in some time b we can get to some possible world w_3 from when we're in w_2 and in time a we can get to some possible world w_2 when we're in w_1, then in time $a + b$ we can get to w_3 from w_1

This axiom is akin to us regarding a string of exclusively possible or exclusively necessary modal operators as just one possible or necessary modal operator, respectively; we regard then regard sum of times as the amount of time it takes to bring about p (again, being necessary or possible, respectively).

Modal Axiom 5: Symmetry

$$\Diamond_s \Box_t p \to \Box_t p \tag{6}$$

We can similarly understand that our modal logic is symmetric; this is valid intuitively from the behavior of the demon. We further abbreviate our treatment for brevity. In terms of the demon, we note that in our semantics $\Diamond_s p$ means that it is possible for the demon to bring about conclusion p via its choices in time s. We can also say $\Box_t p$ means that it is necessary for the demon to bring about p in time t. If it is possible in time s that it is necessary in time s to bring about p, this is equivalent in our semantics to saying that it is necessary in time t to bring about conclusion p. In other words, we can disregard would could have possibly happened in time s from the demon's behavior and only regard what was necessary in time t for the demon to do in order to bring about p.

As another example, consider our reduction of $\Diamond_t \neg \Diamond_t p$ to $\neg \Diamond_t p$ in our proof for deriving a defended reasoner in Section 5. While the intuitive English reading ("It's possible that it's not possible that p") does not seem equivalent to this reduction ("It's not possible that p), it is in fact valid for our semantics. Think of this in terms of the demon. If p cannot be brought about in time t in expectation (where t is a reasonable upper bound on compute time), then that's the end of it; it doesn't matter which operators come before it (any number of $\Diamond_t$ or $\Box_t$). Adding possibility or necessity before that condition doesn't change that fact that it, for that upper bound t, it is not possible to bring about p.

This axiom is akin to us just regarding the rightmost modal operator when we have a mix of modal operators applied iteratively; we can disregard what was possible or necessary in the time prior to the rightmost operator, and say that what we can say about a sentence p (whether it is possible or necessary) just relates to how much time the last operator required to bring about p.

Derived axioms

We can similarly derive other axioms of our indexed modal logic, form the axioms above. Notably,

$\Box_t$ **distributes over** $\land$

$\Box_t(p \land q) \to (\Box_t p \land \Box_t q)$ ($\Box_t$ *distributes over* $\land$)
        Inner proof 1
                $p \land q$
                    p
           $(p \land q) \to p$
      $\Box_t((p \land q) \to p)$ (*Necessitation*)

30

$$\square_t(p \wedge q) \to \square_t p \qquad\qquad (\textit{Distribution})$$
$$\square_t p \qquad\qquad (\textit{By assuming the hypothesis})$$

Inner proof 2

$$p \wedge q$$
$$q$$
$$(p \wedge q) \to q$$
$$\square_t((p \wedge q) \to q) \qquad\qquad (\textit{Necessitation})$$
$$\square_t(p \wedge q) \to \square_t q \qquad\qquad (\textit{Distribution})$$
$$\square_t q \qquad\qquad (\textit{By assuming the hypothesis})$$
$$\square_t p \wedge \square_t q \qquad\qquad (\textit{By inner proof 1, inner proof 2, assuming the hypothesis})$$

We can show a similar result for $\lozenge_t$ and $\wedge$, omitted for brevity.

$\lozenge_t$ distributes over $\vee$

$$\lozenge_t(p \vee q) \to (\lozenge_t p \vee \lozenge_t q) \qquad\qquad (\lozenge \ \textit{distributes over} \ \vee)$$
$$\neg\square_t\neg(p \vee q) \to (\lozenge_t p \vee \lozenge_t q) \qquad\qquad (\lozenge_t a \equiv \neg\square_t\neg a)$$
$$\neg\square_t(\neg p \wedge \neg q) \to (\lozenge_t p \vee \lozenge_t q) \qquad\qquad (\neg(a \vee b) \equiv (\neg a \wedge \neg b))$$
$$\neg(\square_t\neg p \wedge \square_t\neg q) \to (\lozenge_t p \vee \lozenge_t q) \qquad\qquad (\square_t \ \textit{distributes over} \ \wedge)$$
$$(\neg\square_t\neg p \vee \neg\square_t\neg q) \to (\lozenge_t p \vee \lozenge_t q) \qquad\qquad (\neg(a \wedge b) \equiv (\neg a \vee \neg b))$$
$$(\lozenge_t p \vee \lozenge_t q) \to (\lozenge_t p \vee \lozenge_t q) \qquad\qquad (\lozenge_t a \equiv \neg\square_t\neg a)$$

We can show a similar result for $\square_t$ and $\vee$, omitted for brevity.

D.2.4 Syntax and Semantics for the Logic of our Belief in EHPO Conclusions

The logic of belief is a type of modal logic, called doxastic logic [35], where the modal operator $\mathcal{B}$ is used to express belief[12] Different types of reasoners can be defined using axioms that involve $\mathcal{B}$ [67].

We can formulate the doxastic logic of belief in Backus-Naur form:

For any atomic proposition P, we define recursively a well-formed formula ϕ as

$$\phi \coloneqq P \mid \neg\phi \mid \phi \wedge \phi \mid \mathcal{B}\phi \tag{7}$$

where $\mathcal{B}$ means "It is believed that ϕ". We interpret this recursively where p is the base case, meaning that ϕ is p if it is an atom, $\neg\phi$ is well-formed if ϕ is well-formed. We can also define $\vee$, $\to$, and $\leftrightarrow$ from $\neg$ and $\wedge$, as in propositional logic.

As stated in the belief logic portion of Section 4, we model a consistent Type 1 reasoner [67], which has access to all of propositional logic, has their beliefs logical closed under *modus ponens*, and does not derive contradictions. In other words, we have the following axioms:

$$\neg(\mathcal{B}p \wedge \mathcal{B}\neg p) \equiv \mathcal{B}p \to \neg\mathcal{B}\neg p$$

which is the consistency axiom,

$$\vdash p \to \mathcal{B}p$$

which is akin to Necessitation above in Section D.1.1 and means that we believe all tautologies, and

$$\mathcal{B}(p \to q) \to (\mathcal{B}p \to \mathcal{B}q)$$

which means that belief distributes over implication. This notably does not include

---

[12]Computer scientists do not tend to distinguish between the logic of knowledge (epistemic) and the logic of belief (doxastic) [63].

$$\mathcal{B}p \to p$$

which essentially means that we do not allow for believing p to entail concluding p. This corresponds to us actually wanting to run hyperparameter optimization before we conclude anything to be true. We do not just want to conclude something to be true based only on *a priori* information. This is akin to picking folkore parameters and concluding they are optimal without running hyperparameter optimization.

D.2.5 Combining Logics

It is a well known result that we can combine modal logics to make a multimodal logic [61]. In particular, we refer the reader to results on *fusion* [68].

For a brief intuition, we are able to combine our EHPO logic with belief logic since we are operating over the same set of possible worlds. The results of running EHPO produce a particular possible world, to which we apply our logic of belief in order to reason about the conclusions drawn in that world.

D.2.6 Our Combined, Multimodal Logic and Expressing Hyperparameter Deception

We develop the following multimodal logic, which we also state in Section 4:

$$\psi := P \mid \neg\psi \mid \psi \wedge \psi \mid \Diamond_t \psi \mid \mathcal{B}\psi$$

D.2.7 Axioms

We give this multimodal logic semantics to express our t-non-deceptiveness axiom, which we repeat below for completeness:

For any formula p,

$$\neg\left(\Diamond_t \mathcal{B}p \wedge \Diamond_t \mathcal{B}\neg p\right)$$

We can similarly express the t-non-deceptiveness axiom:

For any formula p,

$$\Diamond_t \mathcal{B}p \to \neg\Diamond_t \mathcal{B}\neg p$$

We can also express a t-deceptiveness-axiom:

For any formula p,

$$\Diamond_t \mathcal{B}p \wedge \Diamond_t \mathcal{B}\neg p$$

To reiterate, *multimodal* just means that we have multiple different modes of reasoning, in this case our $\Diamond_t$ semantics for the demon doing EHPO and our consistent Type 1 reasoner operator $\mathcal{B}$.

Given a reasonable maximum time budget t, we say that EHPO is t-non-deceptive if it satisfies all of axioms above. Moreover, based on this notion of t-non-deceptiveness, we can express what it means to have a defense to being deceived.

D.2.8 Some notes on strength of belief and belief update

There are potentially interesting connections between our work on defending against hyperparameter deception and belief update [19]. Notably, one could view our notion of skeptical belief as related to work done on "strength of belief" and belief update, or dynamic doxastic logic [43, 63, 71]. Instead of picking an EHPO runtime *a priori* and then running a defended EHPO and at the end evaluating whether or not we believe the conclusions we draw, we could iteratively update and test our belief and terminate if a certain belief threshold is met. In such quantitative theories of belief change, the degree of acceptance of a sentence is represented by a numerical value. Those numerical values can be updated in light of new information (so-called "soft" information updates) [2, 70]. Exploring this is out of scope for our work here, but could be an interesting future research direction for how to reason about empirical results that imply inconsistent conclusions.

E Section 5 Appendix: Additional Notes on Defenses

E.1 Proving a defended reasoners

Suppose that we have been drawing conclusions using some "naive" belief operator $\mathcal{B}_n$ (based on a conclusion function $\mathcal{F}_n$) that satisfies the axioms of Section 4.3. We want to use $\mathcal{B}_n$ to construct a new operator $\mathcal{B}_*$, which is guaranteed to be deception-free. One straightforward way to do this is to define the belief operator $\mathcal{B}_*$ such that for any statement p,

$$\mathcal{B}_* p \ \equiv \ \mathcal{B}_n p \ \wedge \ \neg \Diamond_t \mathcal{B}_n \neg p.$$

That is, we conclude p only if both our naive reasoner would have concluded p, and it is impossible for an adversary to get it to conclude $\neg p$ in time t. This enables us to show t-non-deceptiveness, following directly from the axioms in a proof by contradiction: Suppose $\mathcal{B}_*$ can be deceived, i.e. $\Diamond_t \mathcal{B}_* p \wedge \Diamond_t \mathcal{B}_* \neg p$ is True:

		Rule
$\Diamond_t \mathcal{B}_* p$	$\equiv \ \Diamond_t (\mathcal{B}_n p \wedge \neg \Diamond_t \mathcal{B}_n \neg p)$	Applying $\Diamond_t$ to the definition of $\mathcal{B}_* p$ (1)
	$\rightarrow \ \Diamond_t (\neg \Diamond_t \mathcal{B}_n \neg p)$	Reducing a conjunction to either of its terms: $(a \wedge b) \rightarrow b$
	$\rightarrow \ \neg \Diamond_t \mathcal{B}_n \neg p$	Symmetry; dropping all but the right-most operator: $\Diamond_t(\Diamond_t a) \rightarrow \Diamond_t a$

We provide more detail on these transformations than we do in the main text. The first application is simple; we just put parentheses around our definition of $\mathcal{B}_*$, and apply $\Diamond_t$ to it. The second step is also simple. We apply a change to whats inside the parentheses, i.e. just the definition of $\mathcal{B}_*$. Because this is a conjunction, in order for it to be true, both components have to be true. So, we can reduce the conjunction to just it's second term.

The part that is more unfamiliar is the application of Modal Axiom 5 (Symmetry) to reduce the number of $\Diamond_t$ operators. We provide this example above in Section D, where we explain why Modal Axiom 5 holds for our EHPO logic semantics. We reiterate here for clarity:

While the intuitive English reading ("It's possible that it's not possible that p") does not seem equivalent to this reduction ("It's not possible that p), it is in fact valid for our semantics. Think of this in terms of the demon. If p cannot be brought about in time t in expectation (where t is a reasonable upper bound on compute time), then that's the end of it; it doesn't matter which operators come before it (any number of $\Diamond_t$ or $\Box_t$). Adding possibility or necessity before that condition doesn't change that fact that it, for that upper bound t, it is not possible to bring about p.

We then pause to apply our axioms to the right side of the conjunction, $\Diamond_t \mathcal{B}_* \neg p$:

		Rule
$\Diamond_t \mathcal{B}_* \neg p$	$\equiv \ \Diamond_t (\mathcal{B}_n \neg p \wedge \neg \Diamond_t \mathcal{B}_n p)$	Applying $\Diamond_t$ to the definition of $\mathcal{B}_* \neg p$ (1)
	$\rightarrow \ \Diamond_t \mathcal{B}_n \neg p \wedge \Diamond_t \neg \Diamond_t \mathcal{B}_n p$	Distributing $\Diamond_t$ over $\wedge$: $\Diamond_t(a \wedge b) \rightarrow (\Diamond_t a \wedge \Diamond_t b)$
	$\rightarrow \ \Diamond_t \mathcal{B}_n \neg p$	Reducing a conjunction to either of its terms: $(a \wedge b) \rightarrow a$

This transformation is much like the one above. We similarly apply $\Diamond_t$ to the definition of $\mathcal{B}_*$.

We then distribute $\Diamond_t$ over the definition, which holds for our logic since possibility distributes over and. We prove this for our logic in Section D, and provide an intuitive explanation here. If it is possible in time t to bring about a particular formula, then it must also be possible to bring about the sub-conditions that compose that formula in time t. If this were not the case, then we would not be able to satisfy bringing about the whole formula in time t.

Lastly, as in the first example, we reduce the conjunction to one of its terms (this time taking the first, rather than the second).

We now bring both sides of the conjunction back together: $\Diamond_t \mathcal{B}_* p \wedge \Diamond_t \mathcal{B}_* \neg p \ \equiv \ \neg \Diamond_t \mathcal{B}_n \neg p \wedge \Diamond_t \mathcal{B}_n \neg p$. The right-hand side is of the form $\neg a \wedge a$, which must be False. This contradicts our initial assumption that $\mathcal{B}_*$ is t-deceptive (i.e., $\Diamond_t \mathcal{B}_* p \wedge \Diamond_t \mathcal{B}_* \neg p$ is True). Therefore, $\mathcal{B}_*$ is t-non-deceptive.

E.2 Theoretically Validating Defenses to Hyperparameter Deception

We prove Theorem 1:

Theorem 1. *Suppose that the set of allowable hyper-HPs $\mathcal{C}$ of H is constrained, such that any two allowable random-search distributions μ and ν have Renyi-∞-divergence at most a constant, i.e. $D_\infty(\mu\|\nu) \leq \gamma$. The (K,R)-defended random-search EHPO of Definition 7 is guaranteed to be t-non-deceptive if we set $R \geq \sqrt{t\exp(\gamma K)/K} = O(\sqrt{t})$.*

Suppose we are considering HPO via random search [4], in which the set of allowable hyper-hyperparameters contains tuples (μ, M), where μ is a distribution over all possible hyperparameter sets Λ and M is the number of different hyperparameter configuration trials to run. This set S is the Cartesian product of the set of allowable distributions D ($\mu \in D$) and M.

Suppose that for any two allowable distributions $\mu, \nu \in D$ and any event A (a measurable subset of Λ), $\mu(A) \leq e^\gamma \cdot \nu(A)$ (i.e., the Renyi ∞-divergence between any pair of distributions is at most γ). This bounds how much the choice of hyper-hyperparameter can affect the hyperparameters in HPO.

We also suppose we start from a naive reasoner (expressed via the operator $\mathcal{B}_n$), which draws conclusions based on a log with K trials. For this scenario, we are only concerned with the reasoner's conclusions from K-trial logs. We therefore assume w.l.o.g. that the reasoner draws no conclusions unless presented with exactly one log with exactly K trials.

For some constant $R \in \mathbb{N}$, we construct a new reasoner $\mathcal{B}_*$ that does the following: It draws conclusions only from a single log with exactly KR trials (otherwise it concludes nothing). To evaluate a proposition p, it splits the log into R groups of K trials each, evaluates $\mathcal{B}_n$ on p on each of those R groups, and then concludes p only if $\mathcal{B}_n$ also concluded p on all R groups.

Now consider a particular (arbitrary) proposition p. Since $\mathcal{B}_*$ draws conclusions based on only a single log, any strategy σ for the demon is equivalent to one that maintains at most one log at all times (the "best" log it found so far for its purposes, as it can discard the rest).

Let Q be the supremum, taken over all allowable distributions μ, of the probability that running a group of K random search trials using that distribution will result in a log that would convince the $\mathcal{B}_n$ of p. Similarly, let $Q_\neg$ be the supremum, taken over all allowable distributions ν, of the probability that running a group of K trials using that distribution will result in a log that would convince $\mathcal{B}_n$ of $\neg p$.

Observe that Q is the probability of an event in a product distribution of K independent random variables each distributed according to μ, and similarly for $Q_\neg$, and the corresponding events are disjoint. By independent additivity of the Renyi divergence, the Renyi ∞-divergence between these corresponding product measures will be γK. It follows that

$$1 - Q \geq \exp(-\gamma K)Q_\neg$$

and

$$1 - Q_\neg \geq \exp(-\gamma K)Q$$

From here it's fairly easy to conclude that

$$Q + Q_\neg \leq \frac{2}{1 + \exp(-\gamma K)}.$$

Now, an EHPO procedure using random search with KR trials will convince $\mathcal{B}_*$ of p with probability Q^R, since all R independently sampled groups of K trials must "hit" and each hit happens with probability Q. Thus, the expected time it will take the fastest strategy to convince us of p is $Q^{-R} \cdot KR$. Similarly, the fastest strategy to convince us of $\neg p$ takes expected time $Q_\neg^{-R} \cdot KR$.

Suppose now, by way of contradiction, that the t-non-deceptiveness axiom is violated, and there are strategies that can achieve both of these in time at most t. That is,

$$Q^{-R} \cdot KR \leq t \qquad \text{and} \qquad Q_\neg^{-R} \cdot KR \leq t.$$

34

From here, it's fairly easy to conclude that

$$Q + Q_\neg \geq 2 \left(\frac{KR}{t} \right)^{1/R}.$$

Combining this with our conclusion above gives

$$\left(\frac{KR}{t} \right)^{1/R} \leq \frac{1}{1 + \exp(-\gamma K)}.$$

It's clear that we can cause this to be violated by setting R to be large enough. Observe that

$$\frac{1}{1 + \exp(-\gamma K)} \leq \exp(- \exp(-\gamma K)),$$

so

$$\left(\frac{KR}{t} \right)^{1/R} \leq \exp(- \exp(-\gamma K)).$$

Taking the root of both sides gives

$$\left(\frac{KR}{t} \right)^{\frac{1}{R \exp(-\gamma K)}} \leq \frac{1}{e}.$$

To simplify this expression, let β denote

$$\beta = R \exp(-\gamma K).$$

So that

$$\left(\frac{\beta K}{t \exp(-\gamma K)} \right)^{1/\beta} \leq \frac{1}{e}.$$

Finally, we set R such that

$$\beta = \sqrt{\frac{t \exp(-\gamma K)}{K}}.$$

To give

$$\left(\frac{1}{\beta} \right)^{1/\beta} \leq \frac{1}{e}.$$

But this is impossible, as the minimum of x^x occurs above $1/e$. This setting of R gives

$$R = \beta \exp(\gamma K) = \sqrt{\frac{t \exp(\gamma K)}{K}} = O(\sqrt{t}).$$

This shows that, for this task, if we run our constructed EHPO with $R = O(\sqrt{t})$ assigned in this way, it will be guaranteed to be t-non-deceptive.

E.3 Defense Experiments

In this section we provide more information about the implementation of a random-search-based defense to hyperparameter deception in Wilson et al. [72], which we discuss in Section 5.

E.3.1 Our Implemented Defense Algorithm

The defense we implement in our experiments is a bit different than what we describe in our theoretical results in Section 5. In particular, in practice it is easier to implement subsampling rather than resampling.

Protocol of Selecting Hyper-HPs. As partially illustrated in Figure 1 and elaborated on in Appendix B, Wilson et al. [72]'s choice of hyper-HPs does not capture the space where Adam effectively simulates Heavy Ball. In Wilson et al. [72]i, Adam-specific HPs like numerical variable $\epsilon = 10^{-8}$ [42] are treated as constants, leading to a biased HP-search space.

In contrast, we select the hyper-HPs of ϵ following a dynamic searching protocol:

Inspired by [11], we start from a wide range $\epsilon \in [10^{-12}, 10^{12}]$ as a wide search space. We iteratively select powers-of-10 grids that are uniformly spaced in the logarithmic scale of the current range. For instance, the selected grids for the prior range would be $\{10^{-12}, 10^{-11}, \cdots, 10^{11}, 10^{12}\}$. We perform a single run on each grid selected, and shrink the range towards grids where the best performance are achieved. The shrinkage follows the policy of either $\times 10$ to the lower boundary or $\times 0.1$ to the upper boundary. For example, for the prior range, we found the best performance is achieved on grid 10^{11}, so we multiply the lower boundary 10^{-12} with 10 and shift the range to $\epsilon \in [10^{-11}, 10^{12}]$. Our protocol terminates with $\epsilon \in [10^{10}, 10^{12}]$ as the final hyper-HPs that we use for our defended random search EHPO.

Scaling Learning Rate η. Note that directly applying the hyper-HP of $\epsilon \in [10^{10}, 10^{12}]$ to Adam would lead to extremely slow convergence, since essentially large ϵ indicates a small effective learning rate η. Similar to Choi et al. [11], we explore a shifted hyper-HPs for the η, scaled proportionally with ϵ. Specifically, note that a large ϵ would make the update of Adam approach the update rule in the Heavy Ball method; for any randomly selected $\epsilon \in [10^{10}, 10^{12}]$, we perform the random search of η/ϵ instead of η itself in the space of $[0.5, 0.7]$, which is the search space of HB's learning rate shown in [72].

E.3.2 Experimental setup

We follow the setup from [72], where the details are specified in Section B.2.1.

E.3.3 Code

The code for running these experiments can be found at `https://github.com/pasta41/deception`.

E.3.4 Associated results and logs

In line with our notion of a log, we provide heatmaps of our logs in Figures 13, 14 and that correspond with our results in Section 5. We note that the performance of Heavy Ball for some random seeds is very bad (e.g., 10% test accuracy). The performance varies widely – also nearing 92% for different random seeds. We affirm that this is the search space that yields the best results for Heavy Ball (92%).

The results for Heavy Ball exhibit large variance. This illustrates a strength of our defensed: it actually helps with robustness against potentially making the wrong conclusion about Heavy Ball's performance (more generally), due to not making conclusions off of a single result (and perhaps using a random seed for which performance is particularly bad). We make a different claim about relative algorithm performance than Wilson et al. [72] about Heavy Ball (i.e., we do not claim that it is better than Adam); but we do not reach this conclusion for the wrong reason (i.e., that we got one bad Heavy Ball result for a particular random seed).

Column 1

Random Seed	LR	Acc.
1	1.9e-01	90.46
	2.3e-01	90.81
3	5.9e-01	91.67
	6.4e-01	91.56
5	1.8e-01	90.77
	5.4e-01	91.79
7	9.0e-01	91.90
	1.2e-01	90.14
9	9.0e-01	91.87
	6.7e-01	92.05
11	9.5e-01	91.89
	8.2e-02	88.89
13	2.8e-01	91.24
	6.4e-01	92.05
15	4.8e-01	91.76
	1.2e-01	89.68
17	4.0e-01	91.19
	8.4e-01	92.10
19	2.7e-01	91.55
	6.4e-01	91.85
21	4.1e-01	91.68
	2.6e-01	90.81
23	7.1e-01	91.74
	1.1e-01	89.65
25	3.5e-01	91.64
	3.6e-01	91.66
27	4.5e-02	86.66
	5.6e-01	91.68
29	7.9e-01	92.00
	8.1e-01	92.00
31	5.4e-01	91.70
	5.1e-01	91.71
33	4.1e-02	86.66
	2.2e-01	90.96
35	4.9e-01	91.78
	2.8e-01	90.73
37	9.9e-01	91.90
	6.8e-01	91.88
39	1.4e-02	82.45
	4.6e-01	91.45
41	6.1e-01	91.81
	6.4e-01	91.92
43	4.5e-01	91.74
	4.3e-01	91.60
45	3.1e-01	91.23
	1.7e-01	90.61
47	4.5e-01	91.73
	7.4e-01	91.72
49	5.6e-01	91.85
	8.2e-01	91.68
51	2.7e-01	90.94
	9.7e-02	89.55
53	6.6e-01	91.78
	1.6e-01	90.45
55	9.4e-01	92.07
	1.9e-01	90.80
57	4.0e-01	91.38
	3.0e-01	91.19
59	3.5e-01	91.32
	8.6e-01	92.08
61	6.4e-01	92.03
	3.8e-01	91.33
63	4.6e-01	91.20
	7.6e-01	92.04
65	1.8e-01	90.87
	5.8e-01	91.55
67	6.9e-01	92.00

Column 2

Random Seed	LR	Acc.
68	6.8e-01	91.66
	4.7e-01	91.49
70	4.8e-01	91.65
	4.2e-01	91.61
72	8.1e-02	88.63
	6.4e-02	87.60
74	4.9e-01	91.42
	7.0e-01	91.64
76	5.1e-01	91.47
	3.8e-02	86.44
78	4.1e-02	87.10
	7.9e-01	92.23
80	1.4e-01	89.96
	7.6e-02	88.59
82	2.1e-02	84.15
	7.4e-01	92.08
84	9.4e-01	92.17
	3.6e-02	86.23
86	2.3e-01	91.04
	8.4e-01	91.78
88	6.1e-01	91.78
	5.5e-01	91.71
90	4.6e-01	91.45
	4.7e-01	91.99
92	8.0e-01	91.94
	7.3e-01	91.99
94	2.1e-01	90.67
	5.0e-02	87.40
96	7.8e-01	91.80
	2.2e-01	90.77
98	6.6e-01	91.99
	9.5e-01	91.88
100	6.6e-02	88.26
	5.2e-01	91.57
102	1.6e-01	90.61
	8.5e-01	91.98
104	6.8e-01	91.74
	1.5e-01	90.12
106	6.9e-02	87.87
	9.4e-01	91.84
108	9.0e-01	92.33
	8.1e-01	91.76
110	9.1e-02	88.68
	4.8e-01	91.39
112	5.6e-01	91.97
	7.2e-01	92.02
114	9.2e-01	92.09
	9.3e-02	89.16
116	4.8e-01	91.80
	4.0e-01	91.34
118	3.2e-01	91.76
	5.4e-01	91.43
120	1.3e-01	90.13
	4.7e-01	91.75
122	2.0e-02	84.09
	5.0e-01	91.87
124	9.5e-01	91.73
	2.4e-01	90.53
126	5.2e-01	91.70
	7.9e-01	91.86
128	5.6e-01	91.70
	1.7e-01	90.73
130	1.3e-01	90.03
	3.9e-01	91.92
132	9.8e-01	91.70
	7.2e-01	91.65
134	6.8e-01	91.93

Column 3

Random Seed	LR	Acc.
135	2.7e-01	91.11
	3.5e-01	91.41
137	8.4e-01	91.81
	7.6e-01	92.03
139	7.0e-01	92.02
	2.1e-01	90.42
141	2.7e-01	90.88
	6.8e-01	91.98
143	1.5e-01	90.31
	9.0e-01	92.15
145	1.4e-01	90.09
	8.5e-01	91.70
147	8.6e-01	91.94
	3.1e-01	91.54
149	2.2e-01	91.12
	2.3e-01	90.90
151	3.8e-02	86.39
	3.4e-01	91.49
153	1.3e-02	81.58
	3.0e-01	91.22
155	9.9e-01	92.19
	4.1e-02	86.74
157	7.8e-01	92.05
	6.5e-01	91.76
159	5.4e-01	91.54
	4.0e-01	91.80
161	7.9e-01	91.77
	3.4e-01	91.80
163	1.0e+00	92.29
	7.0e-01	91.80
165	2.2e-01	90.72
	1.3e-01	89.77
167	4.3e-01	91.64
	2.4e-01	91.02
169	6.7e-02	87.79
	2.0e-03	67.96
171	4.1e-01	91.74
	1.0e+00	92.08
173	1.4e-01	90.12
	1.1e-01	89.55
175	2.2e-01	91.39
	9.7e-02	89.70
177	9.0e-01	91.94
	3.1e-01	91.32
179	6.1e-01	91.95
	9.7e-02	89.55
181	8.9e-01	91.98
	8.4e-01	91.67
183	7.7e-01	92.00
	9.3e-01	91.72
185	1.4e-01	90.35
	4.6e-01	91.49
187	4.2e-01	91.53
	8.9e-02	88.99
189	4.5e-01	91.30
	7.8e-01	91.93
191	7.1e-01	91.62
	4.3e-01	91.47
193	8.3e-02	88.77
	3.7e-01	91.24
195	6.1e-01	91.65
	2.4e-01	91.10
197	5.2e-01	91.70
	2.8e-01	91.10
199	4.1e-01	91.45
	6.3e-01	91.82

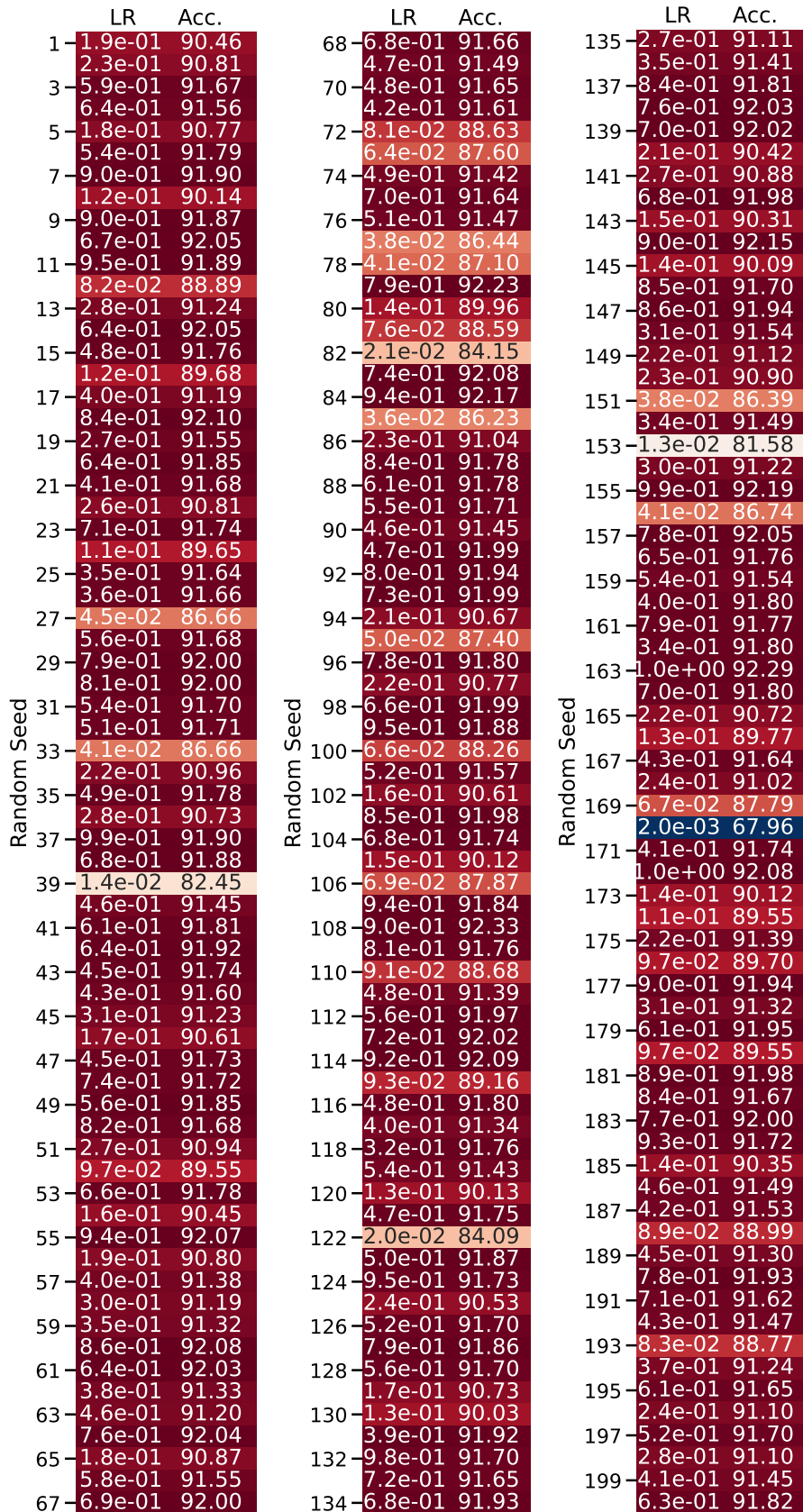Figure 10: Heatmap logs of SGD defended random search. Redder rows indicate higher test accuracy.

Random Seed	LR	Mntm.	Acc.
1	0.77	0.63	92.05
2	0.68	0.72	91.72
3	0.72	0.96	10.00
4	0.93	0.55	10.00
5	0.92	0.54	91.93
6	0.18	0.92	91.59
7	0.96	0.93	10.00
8	0.24	0.85	91.70
9	0.78	0.90	10.00
10	0.58	0.96	10.00
11	0.44	0.99	10.00
12	0.91	0.73	10.00
13	0.49	0.62	91.93
14	0.52	0.84	54.31
15	0.40	0.55	91.97
16	0.72	0.71	91.70
17	0.48	0.71	91.86
18	0.29	0.87	91.82
19	0.99	0.99	10.00
20	0.58	0.93	10.00
21	0.21	0.53	91.10
22	0.25	0.89	91.55
23	0.40	0.76	91.85
24	0.60	0.69	91.96
25	0.52	0.84	10.00
26	0.47	0.74	92.24
27	0.73	0.96	10.00
28	0.96	0.79	10.00
29	0.72	0.54	91.80
30	0.94	0.61	10.00
31	0.20	0.67	91.52
32	0.91	0.84	10.00
33	0.79	0.71	10.00
34	0.15	0.83	91.87
35	0.25	0.78	92.28
36	0.32	0.89	91.01
37	0.38	0.53	91.99
38	0.11	0.53	90.97
39	0.34	0.86	91.59
40	0.46	0.96	10.00
41	0.45	0.82	91.45
42	0.48	0.60	92.07
43	0.51	0.73	92.05
44	0.27	0.60	91.88
45	0.68	0.89	10.00
46	0.72	0.74	90.89
47	0.63	0.96	10.00
48	0.25	0.90	91.24
49	0.07	0.89	92.06
50	0.65	0.61	91.84
51	0.65	0.66	10.00
52	0.04	0.77	90.96
53	0.49	0.95	10.00
54	0.37	0.67	92.22
55	0.48	0.69	92.35
56	0.06	0.67	90.71
57	0.62	0.91	10.00
58	0.90	0.60	10.00
59	0.75	0.52	92.11
60	0.16	0.89	91.77
61	0.76	0.72	10.00
62	0.58	0.84	10.00
63	0.39	0.73	92.19
64	0.73	0.74	91.22
65	0.40	0.92	10.00
66	0.08	0.94	91.74
67	0.99	0.83	10.00
68	0.38	0.62	92.07
69	0.41	0.93	10.00
70	0.90	0.85	10.00
71	0.51	0.60	91.88
72	0.34	0.62	92.00
73	0.88	0.55	10.00
74	0.85	0.98	10.00
75	0.63	0.87	10.00
76	0.40	0.88	28.27
77	0.84	0.87	10.00
78	0.70	0.74	91.70
79	0.60	0.52	91.93
80	0.38	0.74	91.88
81	0.21	0.58	91.71
82	0.66	0.81	28.87
83	0.61	0.73	91.72
84	0.40	0.59	92.11
85	0.59	0.74	91.69
86	0.39	0.94	10.00
87	0.53	0.66	91.91
88	0.54	0.85	10.00
89	0.42	0.53	92.11
90	0.60	0.57	91.95
91	0.99	0.53	10.00
92	0.58	0.74	92.07
93	0.41	0.64	92.41
94	0.29	0.72	92.18
95	0.44	0.77	91.63
96	0.96	0.95	10.00
97	0.49	0.55	91.78
98	0.29	0.53	92.02
99	0.59	0.94	10.00
100	0.96	0.64	10.00
101	0.10	0.94	91.74
102	0.16	0.88	91.85
103	0.65	0.55	91.83
104	0.65	0.66	92.06
105	0.23	0.95	26.17
106	0.63	0.96	10.00
107	0.44	0.85	90.81
108	0.89	0.61	10.00
109	0.30	0.90	89.66
110	0.41	0.70	92.44
111	0.56	0.78	91.80
112	0.15	0.88	92.04
113	0.30	0.81	91.98
114	0.42	0.80	92.03
115	0.33	0.67	91.81
116	0.24	0.94	75.93
117	0.62	0.94	10.00
118	0.93	0.59	10.00
119	0.35	0.77	92.02
120	0.85	0.69	10.00
121	0.68	0.97	10.00
122	0.33	0.80	91.87
123	0.01	0.96	91.14
124	0.46	0.70	92.18
125	0.55	0.57	92.21
126	0.55	0.76	91.50
127	0.27	0.57	91.52
128	0.31	0.74	92.17
129	0.03	0.87	91.04
130	0.80	0.74	10.00
131	0.19	0.68	91.82
132	0.80	0.98	10.00
133	0.20	0.92	90.90
134	0.88	0.56	91.98
135	0.04	0.93	91.74
136	0.42	0.95	10.00
137	0.32	0.84	91.67
138	0.11	0.64	91.34
139	0.00	0.74	81.60
140	0.44	0.59	92.04
141	0.62	0.62	92.05
142	0.37	0.94	10.00
143	0.66	0.64	10.00
144	0.42	0.74	91.82
145	0.44	0.51	91.85
146	0.65	0.97	10.00
147	0.12	0.80	91.93
148	0.41	0.79	92.00
149	0.78	0.75	10.00
150	0.85	0.78	26.81
151	0.23	0.58	91.59
152	0.66	0.88	10.00
153	0.06	0.53	89.91
154	0.03	0.67	89.51
155	0.63	0.81	10.00
156	0.33	0.94	10.00
157	0.33	0.70	92.03
158	0.71	0.74	10.00
159	0.81	0.54	92.09
160	0.99	0.59	91.88
161	0.43	0.85	91.34
162	0.37	0.73	92.09
163	0.55	0.89	10.00
164	0.48	0.88	52.21
165	0.50	0.56	91.82
166	0.78	0.94	10.00
167	0.96	0.51	10.00
168	0.26	0.57	91.78
169	0.48	0.87	80.33
170	0.09	0.91	91.73
171	0.80	0.68	10.00
172	0.97	0.54	92.04
173	0.12	0.78	91.93
174	0.71	0.62	10.00
175	0.77	0.54	91.89
176	0.40	0.87	89.57
177	0.20	0.56	91.43
178	0.64	0.75	91.95
179	0.59	0.74	10.00
180	0.12	0.52	90.99
181	0.24	0.85	91.82
182	0.44	0.61	91.95
183	0.73	0.94	10.00
184	0.37	0.60	91.91
185	0.36	0.70	91.92
186	0.86	0.68	10.00
187	0.99	0.91	10.00
188	0.48	0.76	91.92
189	0.87	0.60	92.06
190	0.91	0.61	10.00
191	0.32	0.53	91.86
192	0.70	0.63	91.93
193	0.34	0.68	91.97
194	0.60	0.72	92.01
195	0.17	0.70	91.67
196	0.49	0.63	92.08
197	0.01	0.61	83.90
198	0.30	0.83	92.02
199	0.52	0.74	91.65
200	0.16	0.83	91.97

Figure 11: Heatmap logs of Heavy Ball (HB) defended random search. Redder rows indicate higher test accuracy.

Seed	LR	Eps.	Acc.		Seed	LR	Eps.	Acc.		Seed	LR	Eps.	Acc.
1	1.7e+11	2.5e+11	92.06		68	1.7e+11	2.8e+11	91.79		135	7.5e+10	1.3e+11	91.58
2	2.0e+11	3.8e+11	91.37		69	7.6e+10	1.1e+11	92.05		136	3.1e+11	5.6e+11	91.29
3	1.3e+11	2.1e+11	91.84		70	1.9e+11	3.1e+11	91.94		137	1.9e+11	2.8e+11	91.71
4	3.8e+11	7.5e+11	91.39		71	2.4e+11	3.6e+11	91.74		138	2.5e+11	4.2e+11	91.64
5	9.8e+10	1.8e+11	91.93		72	4.9e+11	7.1e+11	92.13		139	3.4e+11	5.3e+11	92.02
6	4.4e+11	6.7e+11	91.77		73	5.1e+11	7.3e+11	91.73		140	1.6e+11	2.6e+11	91.77
7	1.8e+11	3.5e+11	91.86		74	6.9e+10	1.3e+11	91.71		141	6.2e+11	9.6e+11	91.54
8	2.7e+11	4.2e+11	91.53		75	1.1e+11	2.3e+11	91.75		142	1.5e+11	2.4e+11	91.60
9	2.6e+11	3.9e+11	91.73		76	8.8e+10	1.4e+11	91.38		143	1.9e+11	3.0e+11	91.38
10	1.8e+11	3.1e+11	91.64		77	4.8e+11	8.7e+11	91.78		144	3.8e+11	6.3e+11	91.72
11	6.3e+11	9.2e+11	91.78		78	8.6e+10	1.5e+11	91.66		145	4.4e+11	6.6e+11	91.15
12	3.0e+11	5.5e+11	91.63		79	5.5e+11	8.8e+11	91.64		146	3.0e+11	4.3e+11	91.68
13	4.2e+11	6.2e+11	91.59		80	1.2e+11	2.3e+11	91.92		147	5.5e+11	9.9e+11	91.21
14	4.5e+11	8.4e+11	91.79		81	2.0e+11	3.1e+11	91.56		148	1.7e+11	2.9e+11	91.75
15	4.6e+11	7.6e+11	91.66		82	7.5e+10	1.1e+11	91.71		149	4.8e+11	8.6e+11	91.74
16	3.8e+11	5.6e+11	91.70		83	9.3e+10	1.5e+11	91.83		150	7.2e+10	1.3e+11	91.47
17	7.1e+10	1.1e+11	91.89		84	4.4e+11	7.4e+11	91.46		151	8.2e+10	1.4e+11	91.30
18	3.2e+11	5.0e+11	91.85		85	3.8e+11	6.1e+11	92.02		152	6.0e+10	1.1e+11	91.73
19	5.6e+10	1.0e+11	91.54		86	1.7e+11	3.0e+11	91.62		153	2.4e+11	4.7e+11	91.51
20	6.0e+10	1.2e+11	91.40		87	9.2e+10	1.8e+11	91.64		154	4.1e+11	7.3e+11	91.65
21	3.6e+11	5.9e+11	91.55		88	1.8e+11	3.3e+11	91.83		155	3.0e+11	5.8e+11	91.38
22	5.6e+11	8.7e+11	91.67		89	5.2e+11	9.4e+11	91.55		156	4.1e+11	7.3e+11	91.57
23	2.0e+11	2.9e+11	92.00		90	9.1e+10	1.4e+11	91.80		157	1.8e+11	3.0e+11	91.08
24	1.8e+11	3.0e+11	91.81		91	1.8e+11	2.8e+11	91.38		158	2.1e+11	3.5e+11	91.69
25	2.9e+11	4.6e+11	91.45		92	3.7e+11	6.9e+11	91.54		159	5.8e+11	9.7e+11	91.67
26	1.9e+11	3.5e+11	91.30		93	1.9e+11	2.9e+11	91.70		160	1.5e+11	2.5e+11	91.48
27	3.8e+11	7.4e+11	91.74		94	5.0e+11	7.7e+11	91.58		161	1.6e+11	2.8e+11	91.50
28	5.1e+11	7.6e+11	91.64		95	2.2e+11	3.3e+11	91.43		162	6.0e+11	1.0e+12	91.81
29	4.7e+11	6.8e+11	91.39		96	5.9e+11	9.9e+11	91.72		163	8.2e+10	1.2e+11	91.63
30	5.1e+11	7.7e+11	91.37		97	2.8e+11	5.3e+11	91.80		164	4.9e+11	8.3e+11	91.50
31	3.2e+11	5.8e+11	91.48		98	6.3e+11	9.3e+11	91.91		165	1.3e+11	2.1e+11	91.49
32	1.9e+11	3.0e+11	91.72		99	3.0e+11	5.2e+11	91.67		166	4.8e+11	7.2e+11	91.69
33	6.5e+10	1.1e+11	91.53		100	2.0e+11	3.2e+11	91.92		167	4.6e+11	7.2e+11	91.60
34	2.8e+11	4.7e+11	91.88		101	2.1e+11	3.7e+11	91.93		168	1.3e+11	2.0e+11	91.71
35	3.6e+11	5.7e+11	91.63		102	4.1e+11	8.1e+11	91.60		169	3.5e+11	7.0e+11	91.46
36	4.8e+11	8.3e+11	91.55		103	4.5e+11	6.7e+11	91.93		170	4.7e+11	7.5e+11	91.56
37	5.5e+11	8.2e+11	91.84		104	3.8e+11	7.5e+11	91.51		171	4.4e+11	8.0e+11	91.78
38	4.1e+11	7.8e+11	91.56		105	5.3e+11	9.5e+11	91.74		172	3.3e+11	5.8e+11	91.70
39	2.3e+11	4.3e+11	91.90		106	2.6e+11	4.1e+11	92.05		173	5.8e+11	8.8e+11	91.84
40	4.2e+11	8.2e+11	91.44		107	3.4e+11	5.0e+11	91.47		174	2.8e+11	4.9e+11	91.73
41	1.5e+11	2.7e+11	91.61		108	3.0e+11	5.6e+11	91.65		175	1.1e+11	1.9e+11	91.80
42	6.6e+10	1.2e+11	91.53		109	8.6e+10	1.5e+11	91.67		176	4.4e+11	8.7e+11	91.41
43	4.2e+11	7.0e+11	91.98		110	4.6e+11	6.6e+11	91.51		177	9.4e+10	1.5e+11	91.71
44	1.8e+11	3.1e+11	91.58		111	2.3e+11	3.5e+11	91.83		178	1.7e+11	3.0e+11	91.51
45	3.8e+11	5.9e+11	91.80		112	1.9e+11	2.9e+11	91.61		179	1.1e+11	1.6e+11	91.97
46	3.3e+11	5.5e+11	91.76		113	4.2e+11	7.5e+11	91.22		180	1.4e+11	2.1e+11	91.52
47	4.5e+11	6.5e+11	91.56		114	2.4e+11	4.7e+11	91.52		181	9.8e+10	1.9e+11	91.81
48	6.7e+10	1.3e+11	91.99		115	3.2e+11	6.4e+11	91.61		182	6.1e+11	8.8e+11	91.63
49	3.4e+11	6.3e+11	91.44		116	2.2e+11	3.5e+11	92.05		183	3.8e+11	5.7e+11	91.58
50	1.5e+11	2.1e+11	91.42		117	6.2e+11	9.2e+11	91.74		184	5.8e+11	9.3e+11	91.88
51	2.5e+11	4.5e+11	91.54		118	4.3e+11	7.9e+11	91.33		185	2.0e+11	3.1e+11	91.50
52	1.8e+11	3.3e+11	91.94		119	5.7e+11	8.5e+11	91.30		186	5.6e+11	9.5e+11	91.98
53	2.3e+11	4.2e+11	91.62		120	4.6e+11	7.8e+11	91.77		187	4.5e+11	7.9e+11	91.29
54	4.4e+11	8.4e+11	91.69		121	1.3e+11	2.4e+11	91.33		188	4.7e+11	7.0e+11	92.02
55	2.6e+11	4.1e+11	92.14		122	2.2e+11	4.1e+11	91.86		189	1.0e+11	1.8e+11	91.70
56	6.6e+11	9.7e+11	91.63		123	6.5e+10	1.0e+11	91.92		190	1.6e+11	2.9e+11	91.50
57	1.1e+11	2.1e+11	91.69		124	5.0e+11	8.8e+11	91.74		191	3.1e+11	4.9e+11	91.81
58	4.6e+11	9.0e+11	91.77		125	5.4e+11	1.0e+11	91.42		192	1.5e+11	2.6e+11	92.02
59	3.0e+11	4.8e+11	91.97		126	4.6e+11	8.5e+11	91.81		193	1.2e+11	2.1e+11	91.51
60	1.6e+11	2.9e+11	91.88		127	8.8e+10	1.3e+11	91.42		194	4.4e+11	7.7e+11	91.87
61	3.7e+11	6.4e+11	91.64		128	3.1e+11	5.2e+11	91.76		195	3.6e+11	5.8e+11	91.48
62	3.1e+11	5.4e+11	91.81		129	5.5e+11	9.3e+11	91.53		196	2.8e+11	4.0e+11	91.89
63	4.7e+11	8.6e+11	91.59		130	5.7e+11	9.8e+11	92.00		197	4.7e+11	9.2e+11	91.83
64	4.1e+11	6.6e+11	91.67		131	4.3e+11	8.4e+11	91.56		198	3.0e+11	5.5e+11	91.73
65	4.5e+11	8.2e+11	91.47		132	4.4e+11	6.8e+11	91.01		199	2.7e+11	4.5e+11	91.69
66	2.5e+11	3.7e+11	91.71		133	5.9e+11	9.4e+11	91.52		200	1.7e+11	3.1e+11	91.70
67	1.1e+11	1.9e+11	91.80		134	4.3e+11	7.3e+11	91.44					

Figure 12: Heatmap logs of Adam defended random search. Redder rows indicate higher test accuracy.

Random Seed	Adam	SGD	Random Seed	Adam	SGD	Random Seed	Adam	SGD	Random Seed	Adam	SGD
1	92.06	90.46	51	91.54	90.94	101	91.93	91.57	151	91.30	86.39
	91.37	90.81		91.94	89.55		91.60	90.61		91.73	91.49
3	91.84	91.67	53	91.62	91.78	103	91.93	91.98	153	91.51	81.58
	91.39	91.56		91.69	90.45		91.51	91.74		91.65	91.22
5	91.93	90.77	55	92.14	92.07	105	91.74	90.12	155	91.38	92.19
	91.77	91.79		91.63	90.80		92.05	87.87		91.57	86.74
7	91.86	91.90	57	91.69	91.38	107	91.47	91.84	157	91.08	92.05
	91.53	90.14		91.77	91.19		91.65	92.33		91.69	91.76
9	91.73	91.87	59	91.97	91.32	109	91.67	91.76	159	91.67	91.54
	91.64	92.05		91.88	92.08		91.51	88.68		91.48	91.80
11	91.78	91.89	61	91.64	92.03	111	91.83	91.39	161	91.50	91.77
	91.63	88.89		91.81	91.33		91.61	91.97		91.81	91.80
13	91.59	91.24	63	91.59	91.20	113	91.22	92.02	163	91.63	92.29
	91.79	92.05		91.67	92.04		91.52	92.09		91.50	91.80
15	91.66	91.76	65	91.47	90.87	115	91.61	89.16	165	91.49	90.72
	91.70	89.68		91.71	91.55		92.05	91.80		91.69	89.77
17	91.89	91.19	67	91.80	92.00	117	91.74	91.34	167	91.60	91.64
	91.85	92.10		91.79	91.66		91.33	91.76		91.71	91.02
19	91.54	91.55	69	92.05	91.49	119	91.30	91.43	169	91.46	87.79
	91.40	91.85		91.94	91.65		91.77	90.13		91.56	67.96
21	91.55	91.68	71	91.74	91.61	121	91.33	91.75	171	91.78	91.74
	91.67	90.81		92.13	88.63		91.86	84.09		91.70	92.08
23	92.00	91.74	73	91.73	87.60	123	91.92	91.87	173	91.84	90.12
	91.81	89.65		91.71	91.42		91.74	91.73		91.73	89.55
25	91.45	91.64	75	91.75	91.64	125	91.42	90.53	175	91.80	91.39
	91.30	91.66		91.38	91.47		91.81	91.70		91.41	89.70
27	91.74	86.66	77	91.78	86.44	127	91.42	91.86	177	91.71	91.94
	91.64	91.68		91.66	87.10		91.76	91.70		91.51	91.32
29	91.39	92.00	79	91.64	92.23	129	91.53	90.73	179	91.97	91.95
	91.37	92.00		91.92	89.96		92.00	90.03		91.52	89.55
31	91.48	91.70	81	91.56	88.59	131	91.56	91.92	181	91.81	91.98
	91.72	91.71		91.71	84.15		91.01	91.70		91.63	91.67
33	91.53	86.66	83	91.83	92.08	133	91.52	91.65	183	91.58	92.00
	91.88	90.96		91.46	92.17		91.44	91.93		91.88	91.72
35	91.63	91.78	85	92.02	86.23	135	91.58	91.11	185	91.50	90.35
	91.55	90.73		91.62	91.04		91.29	91.41		91.98	91.49
37	91.84	91.90	87	91.64	91.78	137	91.71	91.81	187	91.29	91.53
	91.56	91.88		91.83	91.78		91.64	92.03		92.02	88.99
39	91.90	82.45	89	91.55	91.71	139	92.02	92.02	189	91.70	91.30
	91.44	91.45		91.80	91.45		91.77	90.42		91.50	91.93
41	91.61	91.81	91	91.38	91.99	141	91.54	90.88	191	91.81	91.62
	91.53	91.92		91.54	91.94		91.60	91.98		92.02	91.47
43	91.98	91.74	93	91.70	91.99	143	91.38	90.31	193	91.51	88.77
	91.58	91.60		91.58	90.67		91.72	92.15		91.87	91.24
45	91.80	91.23	95	91.43	87.40	145	91.15	90.09	195	91.48	91.65
	91.76	90.61		91.72	91.80		91.68	91.70		91.89	91.10
47	91.56	91.73	97	91.80	90.77	147	91.21	91.94	197	91.83	91.70
	91.99	91.72		91.91	91.99		91.75	91.54		91.73	91.10
49	91.44	91.85	99	91.67	91.88	149	91.74	91.12	199	91.69	91.45
	91.42	91.68		91.92	88.26		91.47	90.90		91.70	91.82

Figure 13: Heatmap logs of test accuracy of VGG-16 on CIFAR-10 for Adam vs. SGD using our defended random search EHPO. Red indicates higher test accuracy for the given random seed.

Test Accuracy — VGG-16 on CIFAR-10, Adam vs. Heavy Ball (HB)

Random Seed	Adam	HB	Random Seed	Adam	HB	Random Seed	Adam	HB	Random Seed	Adam	HB
1	92.06	92.05	51	91.54	10.00	101	91.93	91.74	151	91.30	91.59
	91.37	91.72		91.94	90.96		91.60	91.85		91.73	10.00
3	91.84	10.00	53	91.62	10.00	103	91.93	91.83	153	91.51	89.91
	91.39	10.00		91.69	92.22		91.51	92.06		91.65	89.51
5	91.93	91.93	55	92.14	92.35	105	91.74	26.17	155	91.38	10.00
	91.77	91.59		91.63	90.71		92.05	10.00		91.57	10.00
7	91.86	10.00	57	91.69	10.00	107	91.47	90.81	157	91.08	92.03
	91.53	91.70		91.77	10.00		91.65	10.00		91.69	10.00
9	91.73	10.00	59	91.97	92.11	109	91.67	89.66	159	91.67	92.09
	91.64	10.00		91.88	91.77		91.51	92.44		91.48	91.88
11	91.78	10.00	61	91.64	10.00	111	91.83	91.80	161	91.50	91.34
	91.63	10.00		91.81	10.00		91.61	92.04		91.81	92.09
13	91.59	91.93	63	91.59	92.19	113	91.22	91.98	163	91.63	10.00
	91.79	54.31		91.67	91.22		91.52	92.03		91.50	52.21
15	91.66	91.97	65	91.47	10.00	115	91.61	91.81	165	91.49	91.82
	91.70	91.70		91.71	91.74		92.05	75.93		91.69	10.00
17	91.89	91.86	67	91.80	10.00	117	91.74	10.00	167	91.60	10.00
	91.85	91.82		91.79	92.07		91.33	10.00		91.71	91.78
19	91.54	10.00	69	92.05	10.00	119	91.30	92.02	169	91.46	80.33
	91.40	10.00		91.94	10.00		91.77	10.00		91.56	91.73
21	91.55	91.10	71	91.74	91.88	121	91.33	10.00	171	91.78	10.00
	91.67	91.55		92.13	92.00		91.86	91.87		91.70	92.04
23	92.00	91.85	73	91.73	10.00	123	91.92	91.14	173	91.84	91.93
	91.81	91.96		91.71	10.00		91.74	92.18		91.73	10.00
25	91.45	10.00	75	91.75	10.00	125	91.42	92.21	175	91.80	91.89
	91.30	92.24		91.38	28.27		91.81	91.50		91.41	89.57
27	91.74	10.00	77	91.78	10.00	127	91.42	91.52	177	91.71	91.43
	91.64	10.00		91.66	91.70		91.76	92.17		91.51	91.95
29	91.39	91.80	79	91.64	91.93	129	91.53	91.04	179	91.97	10.00
	91.37	10.00		91.92	91.88		92.00	10.00		91.52	90.99
31	91.48	91.52	81	91.56	91.71	131	91.56	91.82	181	91.81	91.82
	91.72	10.00		91.71	28.87		91.01	10.00		91.63	91.95
33	91.53	10.00	83	91.83	91.72	133	91.52	90.90	183	91.58	10.00
	91.88	91.87		91.46	92.11		91.44	91.98		91.88	91.91
35	91.63	92.28	85	92.02	91.69	135	91.58	91.74	185	91.50	91.92
	91.55	91.01		91.62	10.00		91.29	10.00		91.98	10.00
37	91.84	91.99	87	91.64	91.91	137	91.71	91.67	187	91.29	10.00
	91.56	90.97		91.83	10.00		91.64	91.34		92.02	91.92
39	91.90	91.59	89	91.55	92.11	139	92.02	81.60	189	91.70	92.06
	91.44	10.00		91.80	91.95		91.77	92.04		91.50	10.00
41	91.61	91.45	91	91.38	10.00	141	91.54	92.05	191	91.81	91.86
	91.53	92.07		91.54	92.07		91.60	10.00		92.02	91.93
43	91.98	92.05	93	91.70	92.41	143	91.38	10.00	193	91.51	91.97
	91.58	91.88		91.58	92.18		91.72	91.82		91.87	92.01
45	91.80	10.00	95	91.43	91.63	145	91.15	91.85	195	91.48	91.67
	91.76	90.89		91.72	10.00		91.68	10.00		91.89	92.08
47	91.56	10.00	97	91.80	91.78	147	91.21	91.93	197	91.83	83.90
	91.99	91.24		91.91	92.02		91.75	92.00		91.73	92.02
49	91.44	92.06	99	91.67	10.00	149	91.74	10.00	199	91.69	91.65
	91.42	91.84		91.92	10.00		91.47	26.81		91.70	91.97

Figure 14: Heatmap logs of test accuracy of VGG-16 on CIFAR-10 for Adam vs. Heavy Ball (HB) using our defended random search EHPO. Red indicates higher test accuracy for the given random seed.

41

F   Section 6 Appendix: Additional Notes on Conclusion

F.1   Additional Practical Takeaways

In our conclusion in Section 6, we note the following practical takeaways:

- **Researchers should have their own notion of skepticism, appropriate to their specific task.** There is no one-size-fits-all defense solution. Our results are ***broad insights*** about defended EHPO: A defended EHPO is ***always possible***, but finding an efficient one will depend on the task.

- **Researchers should make explicit how they choose hyper-HPs.** What is reasonable is ultimately a function of what the ML community accepts. Being explicit, rather than eliding hyper-HP choices, is essential for helping decide what is reasonable. As a heuristic, we recommend setting hyper-HPs such that they include HPs for which the optimizers' performance starts to degrade, as we do above.

- **Avoiding hyperparameter deception is just as important as reproducibility**. We have shown that reproducibility [7, 29, 34, 57, 64] is only part of the story for ensuring reliability. While necessary for guarding against brittle findings, it is not sufficient. We can replicate results—even statistically significant ones—that suggest conclusions that are altogether wrong.

We elaborate here that our defended random search EHPO indicates a particular form of skepticism that may (or may not) be appropriate to different ML tasks. That is, we suggest a defended EHPO, but do not claim that that EHPO is optimal or suited for all tasks. Even though it may not be optimal, the guarantees it affords would translate to other tasks (so long as the assumption is maintained that there is an upper bound on how much the hyper-HPs can control the HPs). So, while we do not necessarily encourage practitioners to use our particular defended EHPO, we do not discourage it either. The main take away is that practitioners should develop their own notion of skepticism (appropriate to their particular task) and be explicit about the assumptions they rely on when selecting hyper-HPs. The way one chooses hyper-HPs should be defensible.

When in doubt, as a heuristic we recommend using a search space that includes where an algorithm's performance starts to degrade (to be assured that a maximum, even if a local one, has been found). We refer to our dynamic two-phase protocol (which we describe in detail in Appendix E.3.1) as an example of how to do this. We first do a broad (but coarse) search. We used grid search for that initial sweep. Random search may be a better choice for some tasks. We were familiar with Wilson et al. [72] (and many have written about it), and felt confident that grid search would capture the space well based on the results that others have also reported on this task. We then used this first sweep to determine which hyper-HPs we should use for our second, finer-grained sweep. We apply our more expensive, defended EHPO for this second sweep, using the hyper-HPs we selected from the first sweep. In other words, we spent a bit of time/our compute budget justifying to ourselves that we were picking reasonable hyperparameters – instead of just picking one grid or range for random search to sample, and hoping that our results would be representative of other search spaces.

F.2   Broader Impact

As we suggest in Section 5, our work can be considered as related to (but orthogonal with) with prior studies on reproducibility as advocating for more robust scientific practices in ML research. In particular, our work complements prior empirical studies that shine a light on reliability issues in ML—issues that relate particularly to traditionally underspecified choices in hyperparameter optimization [11, 48, 65]. In contrast to this prior work, which illustrates the issue with experiments, we provide a theoretical contribution that enables ML practitioners and researchers to defend against unreliable, inconsistent HPO. We provide a theoretically-backed mechanism to promote and facilitate more trustworthy norms and practices in ML research.

More broadly, our work can be understood as a mechanism for dealing with *measurement bias*—the misalignment between what one intends to measure and what they are actually measuring—for overall ML algorithm performance. While alleviating measurement bias is by no means novel to more mature branches of science [28], including other fields of computing [54], until recently it has been under-explored in ML. Beginning in the last couple of years, measurement bias is now coming under increased scrutiny with respect to the origins of empirical gains in ML [22, 53]. In current work, it is often difficult to disentangle whether the concluded measured performance gains are due to properties of the training algorithm or to fortuitous HP selection. Our formalization, rather than

allowing HPO choices to potentially obscure empirical results, provides confidence in the conclusions we can draw about overall algorithm performance.

Our work also highlights how there is a human element, not a just statistical one, to bias in ML pipelines: Practitioners make decisions about HPO that can heavily influence performance (e.g., choice of hyper-hyperparameters). The human element of biasing solution spaces has been discussed in sociotechnical writing [14, 23, 66], in AI [52], in the context of "p-hacking" or "fishing expeditions" for results that fit a desired pattern [25], and was also the focus of Professor Isbell [40]'s NeurIPS 2020 keynote. Formalizing the process for how to draw conclusions from HPO, as we do here, has the potential to alleviate the effects of this type of human bias in ML systems.

Lastly, our insights concerning robustness also extend to growing areas in ML that use learning to guide hyperparemeter selection, such as meta-learning and neural architecture search [8, 17, 38, 39, 75]. While the assisting learning agents in those methods guide choosing hyperparameters for the trained output model, their own hyperparameters tend to be either manually tuned or chosen with more traditional HPO strategies, like grid search [74]. In other words, these processes can exhibit the bias problem discussed above and are therefore potentially subject to hyperparameter deception, which can be mitigated by the work we present here.